20th European Modeling & Simulation Symposium

Calabria

September 17 - 19 2008
Campora S. Giovanni
Italy

International Mediterranean and Latin American Modeling Multiconference

EMSS 2008

IMM

MAS 2008

HMS 2008

Mechanical Department, University of Calabria

M&S Center - Laboratory of Enterprise Solutions

DIPTEM, University of Genoa

Liophant Simulation

International Mediterranean & Latin American Council of Simulation

Autonomous University of Barcelona

Universidad de La Laguna

Edited by:

Agostino Bruzzone

Francesco Longo

Miquel Angel Piera

Rosa M. Aguilar

Claudia Frydman

THE 20TH EUROPEAN MODELING & SIMULATION SYMPOSIUM

SEPTEMBER 17-19 2008
CAMPORA S. GIOVANNI (AMANTEA, CS), ITALY

EMSS
2008

EDITED BY
AGOSTINO BRUZZONE
FRANCESCO LONGO
MIQUEL ANGEL PIERA
ROSA M. AGUILAR
CLAUDIA FRYDMAN

PRINTED IN RENDE (CS), ITALY, SEPTEMBER 2008

© 2008 R. MOSCA, DIPTEM UNIVERSITÀ DI GENOVA

RESPONSIBILITY FOR THE ACCURACY OF ALL STATEMENTS IN EACH PAPER RESTS SOLELY WITH THE AUTHOR(S). STATEMENTS ARE NOT NECESSARILY REPRESENTATIVE OF NOR ENDORSED BY THE DIPTEM, UNIVERSITY OF GENOA. PERMISSION IS GRANTED TO PHOTOCOPY PORTIONS OF THE PUBLICATION FOR PERSONAL USE AND FOR THE USE OF STUDENTS PROVIDING CREDIT IS GIVEN TO THE CONFERENCES AND PUBLICATION. PERMISSION DOES NOT EXTEND TO OTHER TYPES OF REPRODUCTION NOR TO COPYING FOR INCORPORATION INTO COMMERCIAL ADVERTISING NOR FOR ANY OTHER PROFIT – MAKING PURPOSE. OTHER PUBLICATIONS ARE ENCOURAGED TO INCLUDE 300 TO 500 WORD ABSTRACTS OR EXCERPTS FROM ANY PAPER CONTAINED IN THIS BOOK, PROVIDED CREDITS ARE GIVEN TO THE AUTHOR(S) AND THE WORKSHOP.

FOR PERMISSION TO PUBLISH A COMPLETE PAPER WRITE TO: DIPTEM UNIVERSITY OF GENOA, DIRECTOR, VIA OPERA PIA 15, 16145 GENOVA, ITALY. ADDITIONAL COPIES OF THE PROCEEDINGS OF THE *EMSS* ARE AVAILABLE FROM DIPTEM UNIVERSITY OF GENOA, DIRECTOR, VIA OPERA PIA 15, 16145 GENOVA, ITALY.

The 20th European Modeling & Simulation Symposium

September 17-19 2008
Campora S. Giovanni (Amantea), Italy

Organized by

Mechanical Department, University of Calabria

MSC-LES, Modeling & Simulation Center, Laboratory of Enterprise Solutions

DIPTEM - University of Genoa

Liophant Simulation

Sponsored by

Academic, Institutes and Societies Sponsors

IEEE - Institute of Electrical and Electronics Engineers

SCS - Society for Computer Simulation International

MISS - McLeod Institute of Simulation Science

M&SNet - Modeling & Simulation Network

IMCS - International Mediterranean & Latin American Council of Simulation

MSLS - Autonomous University of Barcelona

University of La Laguna

Industry Sponsors

MAST - Management and Advanced Solutions and Technologies

Tonno Callipo

Liquirizie Amarelli

Giulio Barca Prodotti in Pelle

EDITORS

AGOSTINO BRUZZONE
MISS-DIPTEM, UNIVERSITY OF GENOA
VIA MOLINERO 1
17100 SAVONA, ITALY
agostino@itim.unige.it
www.st.itim.unige.it

FRANCESCO LONGO
MSC-LES, MODELING & SIMULATION CENTER – LABORATORY OF ENTERPRISE SOLUTIONS
MECHANICAL DEPARTMENT
UNIVERSITY OF CALABRIA
VIA P. BUCCI – 87036, RENDE (CS), ITALY
flongo@unical.it
www.msc-les.org

MIQUEL ANGEL PIERA
MSLS, DEPARTMENT OF SYSTEM ENGINEERING
UNIVERSITY AUTONOMOUS OF BARCELONA
08193- BELLATERRA- SPAIN
MiquelAngel.Piera@uab.es
www.uab.es

ROSA M. AGUILAR
DEPARTMENT OF SYSTEMS ENGINEERING AND AUTOMATION AND COMPUTER ARCHITECTURE
UNIVERSITY OF LA LAGUNA
PABELLÓN DE GOBIERNO, C/ MOLINOS DE AGUA S/N., 38207, LA LAGUNA, SPAIN
raguilar@ull.es
www.ull.es

CLAUDIA FRYDMAN
LSIS - LABORATOIRE DES SCIENCES DE L'INFORMATION ET DES SYSTÈMES
UNIVERSITÉ AIX-MARSEILLE (U3)
DOMAINE UNIVERSITAIRE DE SAINT-JÉRÔME, AVENUE ESCADRILLE NORMANDIE-NIEMEN, 13397 MARSEILLE CEDEX 20
claudia.frydman@lsis.org
www.lsis.org

THE INTERNATIONAL MEDITERRANEAN AND LATIN AMERICAN MODELING MULTICONFERENCE, I3M 2008

GENERAL CO-CHAIRS

AGOSTINO BRUZZONE, *MISS DIPTEM, UNIVERSITY OF GENOA, ITALY*
MIQUEL ANGEL PIERA, *AUTONOMOUS UNIVERSITY OF BARCELONA, SPAIN*

PROGRAM CHAIR

FRANCESCO LONGO, *MSC-LES, UNIVERSITY OF CALABRIA, ITALY*

THE 20TH EUROPEAN MODELING & SIMULATION SYMPOSIUM, EMSS 2008

GENERAL CO-CHAIRS

FRANCESCO LONGO, *MSC-LES, UNIVERSITY OF CALABRIA, ITALY*
MIQUEL ANGEL PIERA, *AUTONOMOUS UNIVERSITY OF BARCELONA, SPAIN*

PROGRAM CO-CHAIRS

ROSA M. AGUILAR, *UNIVERSITY OF LA LAGUNA, SPAIN*
CLAUDIA FRYDMAN, *LSIS, UNIVERSITY OF AIX-MARSEILLE*

EMSS 2008 INTERNATIONAL PROGRAM COMMITTEE

Michael Affenzeller, *Upper Austrian Univ. of AS, Austria*
Rosa Mª Aguilar, *University of La Laguna, Spain*
Antonio A. Alonso, *IIM-CSIC, Spain*
Tayfur Altiok, *Rutgers University, USA*
Andreas Beham, *Upper Austrian Univ. of AS, Austria*
Ales Belic, *University of Ljubljana, Slovenia*
Enrico Bocca, *Liophant Simulation, Italy*
Fabio Bruno, *University of Calabria, Italy*
Agostino Bruzzone, *University of Genoa, Italy*
Iván Castilla Rodriguez, *University of La Laguna, Spain*
Antonio Cimino, *MSC-LES, University of Calabria, Italy*
Duilio Curcio, *MSC-LES, Italy*
Sarabia Daniel, *University of Valladolid, Spain*
Donald Dudenhoeffer, *Idaho National Laboratory, USA*
Maria Pia Fanti, *Polytechnic of Bari, Italy*
Luigino Filice, *University of Calabria, Italy*
Claudia Frydman, *LSIS, France*
Martin Geuer, *Suretec Systems, Germany*
Witold Jacak, *Upper Austrian Univ. of AS, Austria*
Andras Javor, *Budapest Univ. of Tech. and Ec., Hungary*
Monika Kofler, *Upper Austrian Univ. of AS, Austria*
Gabriel Kronberger, *Upper Austrian Univ. of AS, Austria*
Werner Kurschl, *Upper Austrian Univ. of AS, Austria*
Pasquale Legato, *University of Calabria, Italy*
Francesco Longo, *MSC-LES, University of Calabria, Italy*
Marina Massei, *MAST, Italy*
Yuri Merkuryev, *Riga Technical University, Latvia*
Jean Marc Mercantini, *University of Aix-Marseille III, France*
Giovanni Mirabelli, *University of Calabria, Italy*
Roberto Muñoz González, *University of La Laguna, Spain*
Gasper Music, *University of Ljubljana, Slovenia*
Maurizio Muzzupappa, *University of Calabria, Italy*
Dan Neagu, *Bradford University, UK*
Tudor Niculiu, *University of Bucharest, Romania*
Libero Nigro, *University of Calabria, Italy*
Tuncer Ören, M&SNet, *University of Ottawa, Canada*
Juan Albino Méndez Pérez,*University of La Laguna,Spain*
Patrick Peschlow, *University of Bonn, Germany*
Miquel Angel Piera, *UAB, Spain*
Erika Pinto, *University of Calabria, Italy*
Cesar de Prada, *Universidad de Valladolid, Spain*
Santiago torres Álvarez, *University of La Laguna, Spain*
Stefano Saetta, *University of Perugia, Italy*
Andrea Santoro, *ENEA, Italy*
Alberto Tremori, *MAST, Italy*
Walter Ukovich, *University of Trieste, Italy*
Ion Vaduva, *University of Bucharest, Romania*
Mihai Vasiliu, *University of Bucharest, Romania*
Levent Yilmaz, *Auburn University, USA*
Stefan Wagner, *Upper Austrian Univ. of AS, Austria*
Gabriel Wainer, *Carleton University, Canada*
Stephan Winkler, *Upper Austrian Univ. of AS, Austria*

TRACKS AND WORKSHOP CHAIRS

Workshop on Soft Computing and Modeling & Simulation
Chairs: Witold Jacak, Michael Affenzeller, *Upper Austrian University of Applied Sciences, (Austria)*

Industrial Processes Modelling & Simulation
Chair: Cesar De Prada, *Universidad de Valladolid, (Spain)*

Emergency Simulation
Chair: Donald Dudenhoeffer, *Idaho National Laboratory (USA)*

Supply Chain Risk & Vulnerability
Chair: Jean Marc Mercantini, *University of Aix-Marseille III, France*

Distributed and Parallel Simulation
Chair: Patrick Peschlow, *University of Bonn (Germany)*

Virtual Reality and Visualization
Chairs: Maurizio Muzzupappa, Fabio Bruno, *University of Calabria (Italy)*

Simulation Software and Optimization Tools
Chair: Antonio A. Alonso, *Consejo Superior de Investigaciones Científicas, IIM-CSIC, (Spain)*

Simulation and Artificial Intelligence
Chair: Tudor Niculiu, *University "Politehnica" of Bucharest, (Romania)*

Discrete and Combined Simulation
Chair: Gasper Music, *University of Ljubljana, Slovenia*

Agent Directed Simulation
Chairs: Tuncer Ören, *University of Ottawa (Canada)*; Levent Yilmaz, *Auburn University (USA)*

Petri Nets and DEVS Modelling & Simulation
Chairs: Libero Nigro, *University of Calabria (Italy)*; Gabriel Wainer, *Carleton University (Canada)*

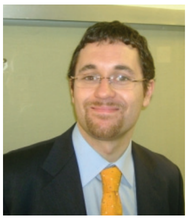
Modelling & Simulation in Business Management
Chair: Iván Castilla, Roberto Muñoz, *Universidad de La Laguna (Spain)*

Simulation based Optimization
Chair: Pasquale Legato, *University of Calabria (Italy)*

Advance in Information and e-Integration in Healthcare Systems and Management
Chairs: Maria Pia Fanti, *Polytechnic of Bari (Italy)*; Walter Ukovich, *University of Trieste (Italy)*

GENERAL CO-CHAIRS' MESSAGE

WELCOME TO EMSS 2008



This 20th EMSS edition asserts the importance of creating a common framework and a high quality International forum for the Simulation Community. The inherent complexity of present industrial, business, design and health care system, the changing mentality in logistics and the constant need to improve competitiveness provoke the necessity of developing better simulation models that could tackle the problem from a global point of view without neglecting any specific sensible aspect of the system's dynamic. Complexity becomes apparent to engineers and managers each time they are asked to take a decision in a context that it is not possible to predict all the consequences of a certain action. Simulation techniques have proved to be useful for examining the performance of different system configurations and/or alternative operating procedures for complex systems. It is widely acknowledged that simulation is a powerful computer-based tool that enables decision-makers in business and industry to improve operational and organizational efficiency.

However, dealing with a good verified and validated simulation model is not an easy task: the experience of the modeler, its capacity to identify the proper abstraction level at which the dynamic of the system should be described, the formalism used to specify the system, and the clarity to discerns what is important and what can be neglected in the model to satisfy the goal of the experiments, are some key aspects that can affect the quality of the solution.

Our main target in the organization of this EMSS'08 has been to create a framework to exchange information, knowledge and the experience acquired by top experts in the relative fields.

This year, and for the fifth time, the European Modeling and Simulation Symposium (EMSS'08) has been organized in the International Mediterranean and Latin American Modeling Multiconference (I3M), born in 2004, as a cooperation among a very wide simulation community that is composed by historical simulation Institutions such as *McLeod Institute of Simulation Science* (MISS, www.simulationscience.org) as well as by new born dynamic organizations such as the *Liophant Simulation* (www.liophant.org), the *Modeling and Simulation Center – Laboratory of Enterprise Solutions* (MSC-LES, www.msc-les.org), and the *Simulation Council of the SCS International: International Mediterranean and Latin America Council of Simulation* (I_M_CS, www.i-m-cs.org).

We wish you a fruitful conference and a pleasant stay in Campora San Giovanni.

Francesco Longo, MSC-LES, University of Calabria, Italy

Miquel Angel Piera, Autonomous University of Barcelona, Spain

ACKNOWLEDGEMENTS

The EMSS 2008 International Program Committee selected the papers for the Conference among many submissions and we expected a very successful event based on their efforts; so we would like to thank all the authors as well as the IPCs and reviewers for their review process.

A special thank to the organizations, institutions and societies that are supporting and technically sponsoring the event: Mechanical Department of University of Calabria, Modeling & Simulation Center – Laboratory of Enterprise Solutions (MSC-LES), DIPTEM - University of Genoa, Liophant Simulation, Autonomous University of Barcelona, McLeod Institute of Simulation Science (MISS), McLeod Modeling & Simulation Network (M&SNet), Institute of Electrical and Electronics Engineers (IEEE), Society for Computer Simulation International (SCS), International Mediterranean & Latin American Council of Simulation (IMCS), the Management and Advanced Solutions and Technologies (MAST), the Tonno Callipo, the Liquirizia Amarelli, the Giulio Barca Prodotti in Pelle. Finally, we would like to thank all the Conference Organization Supporters.

LOCAL ORGANIZATION COMMITTEE

ANTONIO CIMINO, *UNIVERSITY OF CALABRIA, ITALY*
DUILIO CURCIO, *UNIVERSITY OF CALABRIA, ITALY*
FRANCESCO LONGO, *UNIVERSITY OF CALABRIA, ITALY*
GIOVANNI MIRABELLI, *UNIVERSITY OF CALABRIA, ITALY*
ENRICO PAPOFF, *UNIVERSITY OF CALABRIA, ITALY*
ERIKA PINTO, *UNIVERSITY OF CALABRIA, ITALY*
CLAUDIA RUBINO, *CHAMBER OF COMMERCE, CROTONE, ITALY*

Index

Soft Computing and Modeling and Simulation

SELF-LEARNING NAVIGATION MAPS BASED UPON DATA-DRIVEN MODELS USING RECORDED HETEROGENEOUS GPS TRACKS, *Clemens Novak, Barbara Franz, Herwig Mayr, Michal Vesely* 1

BIOMIMETIC MIDDLEWARE FOR WIRELESS SENSOR NETWORKS, *Zenon Chaczko* 7

CARDIOSCOPE SIMULATOR SYSTEM, *Zenon Chaczko, Hany Shehata* 15

A SMART-SHOP SYSTEM - MULTIAGENT SIMULATION SYSTEM FOR MONITORING RETAIL ACTIVITIES, *Zenon Chaczko, Christopher Chiu* 20

SUBPROBLEM SOLVING AND MACHINE PRIORITIZATION IN THE SHIFTING BOTTLENECK PROCEDURE FOR WEIGHTED TARDINESS JOB SHOPS, *Roland Braune* 27

A CLUSTER-BASED OPTIMIZATION APPROACH FOR THE VEHICLE ROUTING PROBLEM WITH TIME WINDOWS, *Michael Bögl* 33

SENSOR NETWORK BASED CONFLICT RESOLUTION IN AUTONOMOUS MULTIAGENT SYSTEMS, *Witold Jacak, Karin Pröll* 39

MODELLING OF IMMUNE FUNCTIONS IN A WIRELESS SENSORS NETWORK, *Jan Nikodem, Ryszard Klempous, Zenon Chaczko* 45

TWO-PHASE SIMULATION OPTIMISATION PROCEDURE WITH APPLICATIONS TO MULTI-ECHELON CYCLIC PLANNING, *Galina Merkuryeva, Liana Napalkova* 51

EFFECTIVE ALLELE PRESERVATION BY OFFSPRING SELECTION: AN EMPIRICAL STUDY FOR THE TSP, *Michael Affenzeller, Stefan Wagner, Stephan Winkler* 59

INCORPORATING PHYSICAL KNOWLEDGE ABOUT THE FORMATION OF NITRIC OXIDES INTO EVOLUTIONARY SYSTEM IDENTIFICATION, *Stephan Winkler, Markus Hirsch, Michael Affenzeller, Luigi del Re, Stefan Wagner* 69

SIMULATION OPTIMIZATION WITH HEURISTICLAB, *Andreas Beham, Michael Affenzeller, Stefan Wagner, Gabriel Kronberger* 75

RECOGNITION OF TRANSITIONS BETWEEN DIFFERENT PHASES OF THE PRODUCT LIFE CYCLE, *Anatoly Sukov* 81

EVOLUTIONARY METAMODELLING OF DISCRETE-EVENT SIMULATION MODELS, *Birkan Can, Gearoid Murphy, Cathal Heavey* 87

DATA MINING VIA DISTRIBUTED GENETIC PROGRAMMING AGENTS, *Gabriel Kronberger, Stephan Winkler, Michael Affenzeller, Stefan Wagner* 95

OPTIMIZATION OF MEDICAL ULTRASOUND TRANSDUCERS WITH SIMULATION AND GENETIC ALGORITHMS, *Monika Kofler, Andreas Beham, Michael Affenzeller, Stefan Wagner* 100

MODELING OF HEURISTIC OPTIMIZATION ALGORITHMS, *Stefan Wagner, Gabriel Kronberger, Andreas Beham, Stephan Winkler, Michael Affenzeller* 106

Virtual Reality and Visualization

A CAD SYSTEM IN AUGMENTED REALITY APPLICATION, *Pier Paolo Valentini, Davide Gattamelata, Eugenio Pezzuti* 112

SPHERICAL HARMONICS ALGORITHM FOR DYNAMIC LIGHT IN REAL TIME, *Lien Muguercia Torres* 119

A NEW HAPTIC-BASED TOOL FOR TRAINING IN MEDICINE, *Tommaso Ingrassia, Vincenzo Nigrelli* 126

A WEB3D APPLICATION FOR THE BIN-PACKING PROBLEM, *Fabio Bruno, Francesco Caruso, Ornella Pisacane, Roberto Musmanno, Maurizio Muzzupappa, Francesco Rende, Giovanni Venuto* 134

MIXED-REALITY ENVIRONMENT BASED ON HAPTICS AND INTERACTIVE SIMULATION FOR PRODUCT DESIGN REVIEW, *Monica Bordegoni, Francesco Ferrise, Marco Ambrogio, Giandomenico Caruso, Fabio Bruno, Francesco Caruso* 140

Industrial Processes Modeling & Simulation

SIMULATION ANALYSES OF NON-KINEMATIC CONICAL ROLL BENDING PROCESS WITH CONICAL ROLLS, *Zhengkun Feng, Henri Champliaud, Thien-My Dao* 146

LIBRARY FOR DYNAMIC SIMULATION OF REVERSE OSMOSIS PLANTS, *Luis G. Palacin, Cesar de Prada, S. Syafiie, Fernando Tadeo* 154

USING AN ADVANCED DISCRETE-EVENT SIMULATION FRAMEWORK TO PRODUCTIVE CAPACITY MANAGEMENT OF A CAR-PARTS FACTORY, *Adrián M. Aguirre, Enrique J. Müller, Sebastián E. Seffino, Carlos A. Méndez* 159

CLASSIC AND ADVANCED MODELS FOR CONTROLLING HVAC SYSTEMS IN A UNIVERSITY BUILDING, *Eladio Sanz, Belén Pérez-Lancho, Pastora Vega, Mario Francisco* 169

MODELING AND EVALUATION OF ALTERNATIVE PRODUCTION SCENARIOS IN THE FIELD OF COMPOSITE MANUFACTURING, *Bernd Scholz-Reiter, Michael Lütjen* 174

ENERGY EFFICIENCY IN A STEEL PLANT USING OPTIMIZATION-SIMULATION, *Ivan Ferretti, Simone Zanoni, Lucio Zavanella* 180

OBJECT ORIENTED LIBRARY FOR DYNAMIC MODELING AND SIMULATION OF SUGAR HOUSES FOR OPERATORS TRAINING, *Rogelio Mazaeda, César de Prada* 188

ANALYSIS AND OPTIMIZATION OF COMPLEX SMALL-LOT PRODUCTION IN NEW MANUFACTURING FACILITIES BASED ON DISCRETE SIMULATION, *Václav Votava, Zdenek Ulrych, Milan Edl, Michal Korecký, Václav Trkovský* 198

A MODEL FOR ENERGY PREDICTIONS OF A HOTEL ROOM, *Adriana Acosta Corzo, Ana Isabel González Santos, Jesús M. Zamarreño Cosme, Víctor Álvarez Castelló* 204

A STOCHASTIC APPROACH TO IMPROVE PERFORMANCE OF PRODUCTION LINES, *Cecilia Zanni, Philippe Bouché* 212

CACSD TOOL FOR SIMULATION AND PERFORMANCE OF MULTI-RATE SAMPLED-DATA SYSTEMS, *Yolanda Cerezo, Ignacio López, Alfredo Cuesta, Luis Grau* 218

EVALUATION AND OPTIMISATION OF MANUFACTURING SYSTEM USING SIMULATION 224

MODELLING AND DESIGN OF EXPERIMENT, *Vittorio Cesarotti, Bruna Di Silvio, Vito Introna, Giovanni Mori*

DATA DRIVEN ADAPTIVE MODEL PREDICTIVE CONTROL WITH CONSTRAINT, *Norhaliza Abdul Wahab, Jonas Balderud, Reza Katebi* 231

PEDESTRAIN CELLULAR AUTOMATA AND INDUSTRIAL PROCESS SIMULATION, *Alan Jolly, Rex Oleson II, David Kaup* 237

Emergency Simulation & Supply Chain Risk and Vulnerability

SIMULATION MODEL OF AIRPORT RUNWAY INCURSIONS, *Luigi Careddu, Francesco Costantino, Giulio Di Gravio* 243

MATHEMATICAL MODELS OF GAS FIRED BOILERS, *Renzo Tosato* 249

TOWARD AN INTEGRATION OF RISK ANALYSIS IN SUPPLY CHAIN ASSESSMENT, *Jean-Claude Hennet, Jean-Marc Mercantini, Isabel Demongodin* 255

SUPPLY CHAIN MANAGEMENT THROUGHT P AND PI CONTROLLERS, *Carlos Andres Garcia, Pedro Balaguer, Ramón Vilanova* 261

FUZZY SIMULATION FOR INFRASTRUCTURE EFFECTS UNCERTAINTY ANALYSIS, *Donald Dudenhoeffer, Milos Manic* 267

Distributed and Parallel Simulation

A PARALLEL PROGRAMMING METHODOLOGY USING COMMUNICATION PATTERNS NAMED CPANS OR COMPOSITION OF PARALLEL OBJECT, *Mario Rossainz López, Manuel I. Capel Tuñón* 274

ANALYSIS OF TIME WARP ON A 32,768 PROCESSOR IBM BLUE GENE/L SUPERCOMPUTER, *Akintayo O. Holder, Christopher D. Carothers* 284

DISCRETE EVENT SIMULATION WITH UNIVERSAL PROGRAMMING LANGUAGES ON MULTICORE PROCESSORS, *Thomas Wiedemann* 293

TEMPLATES FOR DISTRIBUTED AGENT-BASED SIMULATIONS ON A QUASI OPPORTUNISTIC GRID, *Laszlo Gulyas, Walter de Back, Gabor Szemes, Krzysztof Kurowski, Werner Dubitzky, George Kampis* 300

Simulation and Artificial Intelligence

ON-BOARD OPERATIVE ADVICED EXPERT SYSTEMS FOR ONE-SEAT AIRCRAFTS AND STRUCTURE OF THEIR KNOWLEDGE BASES, *Boris Fedunov* 306

MODELING AND SIMULATION OF A NANOSTRUCTURE FOR A SINGLE ELECTRON TECHNOLOGY IMPLEMENTATION, Cristian Ravariu, Adrian Rusu, Ala Bondarciuc, Florina Ravariu, Tudor Niculiu, Florin Babarada, Vlad Bondarciuc 312

SOME ESTHETICAL VALENCES OF THE SPATIAL LOCALITY. TOWARDS ARTIFICIAL ESTHETICS, *Cristian Lupu* 316

HIERARCHICAL INTELLIGENT ANALOG SIMULATION, *Tudor Niculiu, Alexandru Copilau, Cristian Lupu, Anton Manolescu* 323

SIMULATIONS WITH UNIFIED MOSFET MODEL FOR DISTORTIONS ANALYSIS, *Florin Babarada, Adrian Rusu, Tudor Niculiu, Cristian Ravariu, Dragos Vizireanu, Carmen Moldovan, Elena Manea, Camelia Dunare, Madalina Mlak* 333

FAILURE PROCESS SIMULATION OF A COMPONENT-BASED SOFTWARE, *Florentina Suter* 339

EVALUATION OF GA ESTIMATION PARAMETERS IN WIDEBAND PIEZOELECTRIC TRANSDUCERS EMPLOYING A COMPREHENSIVE MODELING TOOL, *Abelardo Ruíz Toledo, David K. Anthony, Antonio Ramos Fernández* 342

A NEURAL MAXIMUM SELECTOR: EXPLICIT PARAMETER SET-UP FOR TIME PERFORMANCE, *Ruxandra L. Costea, Corneliu A. Marinov* 348

OPTIMIZING HINTERLAND TERMINAL OPERATION USING SIMULATION AND NEURAL NETWORKS, *Manfred Gronalt, Thouraya Benna, Martin Posset* 353

QUADRATIC COMPUTATIONAL CIRCUITS FOR VLSI DESIGNS, *Cosmin Popa* 358

Discrete and Combined Simulation

PRACTICAL STABILITY OF POSITIVE FRACTIONAL DISCRETE-TIME LINEAR SYSTEMS, *Tadeusz Kaczorek* 362

ENTITIES WITH COMBINED DISCRETE-CONTINUOUS ATTRIBUTES IN DISCRETE-EVENT-DRIVEN SYSTEMS, *Kristina Dammasch, Graham Horton* 368

SIMULATION MODEL OF A POLYMERIZATION PLANT, *Dejan Gradisar, Vladimir Jovan, Sebastjan Zorzut* 374

TIMED PETRI NET SIMULATION AND RELATED SCHEDULING METHODS: A BRIEF COMPARISON, *Gasper Music* 380

ARTIFICIAL NEURAL NETWORK-BASED CLASSIFICATION OF VECTOR SETS FOR SURFACE INSPECTION, *Michael Gyimesi, Felix Breitenecker, Wolfgang Heidl, Christian Eitzinger* 386

DISTRIBUTED DISCRETE SIMULATION ON THE WEB, *Aman Atri, Felix Breitenecker, Nicole Nagele, Shabnam Tauböck* 392

STRUCTURAL FEATURES IN SIMULATION SYSTEMS – EVOLUTION AND COMPARISON, *Felix Breitenecker, Nikolas Popper, Günther Zauner* 398

INTERNAL/EXTERNAL EVENT – STRUCTURE IN SIMULATORS – CASE STUDIES WITH ARGESIM COMPARISONS, *Felix Breitenecker, Siegried Wassertheurer, Stefan Emrich, Nicolas Popper, Günther Zauner* 408

A PETRI NET APPROACH TO ARGESIM COMPARISON C2 ‘FLEXIBLE ASSEMBLY SYSTEM’ USING THE MATLAB PETRISIMM TOOLBOX, *Thomas Löscher, Felix Breitenecker* 419

NEW APPROACHES OF COMBINED MODELLING AND SIMULATION IN HEALTH CARE SYSTEMS AND MANAGEMENT, *Nikolas Popper, Michael Gyimesi, Günther Zauner, Felix Breitenecker* 425

Petri Nets and DEVS Modeling & Simulation

A PETRI NET MODEL FOR AN ACTIVE DATABASE SIMULATOR, *Joselito Medina-Marín, Xiaoou Li, José Ramón Corona-Armenta, Marco Antonio Montúfar-Benítez, Oscar Montaño-Arango, Aurora Pérez-Rojas* 431

DEGOMS, A SYSTEMATIC WAY FOR TASK MODELLING AND SIMULATION, *Ali Mroue, Jean Caussanel* 438

NET CENTRIC MODELLING AND SIMULATION USING ACTORDEVS, *Franco Cicirelli, Angelo Furfaro, Andrea Giordano, Libero Nigro* 447

TEMPORAL ANALYSIS OF COMPLEX TIME-DEPENDENT SYSTEMS: AN APPROACH BASED ON TIME PETRI NETS, ACTORDEVS AND HLA, *Franco Cicirelli, Angelo Furfaro, Libero Nigro, Francesco Pupo* 455

Modeling & Simulation in Business Management

HYBRID APPROACH FOR MODELING AND CONTROL WAREHOUSE SYSTEMS, *Filippo Sarri, Rinaldo Rinaldi* 463

IMPROVING A PROCESS IN A BRAZILIAN AUTOMOTIVE PLANT APPLYING PROCESS MAPPING, DESIGN OF EXPERIMENTS AND DISCRETE EVENTS SIMULATION, *José Arnaldo Montevechi, Alexandre de Pinho, Fabiano Leal, Fernando Marins, Rafael Costa* 472

AN APPLICATION FOR WEB-BASED MODELING AND SIMULATION, *Yurena García-Hevia, Ivan Castilla, Rosa María Aguilar, Roberto Muñoz* 481

TRANSIENT SIMULATION OF BALANCED BIDDING IN KEYWORD AUCTIONS, *Maurizio Naldi, Giuseppe D'Acquisto* 487

A MARKOV PROCESS FOR REFLECTIVE PETRI NETS, *Lorenzo Capra* 493

ANALYSIS, MODELING AND SIMULATION OF THE INCORPORATION OF EGOVERNMENT IN ADMINISTRATIVE PROCESSES, *Pedro Baquero, Yurena Garcia-Hevia, Rosa María Aguilar* 499

SIMULATION OF TRUST IN CLIENT- WEALTH MANAGEMENT ADVISOR RELATIONSHIPS, *Terry Bossomaier, Russell Standish* 506

ENABLING ADVANCED SIMULATION SCENARIOS WITH NEW SOFTWARE ENGINEERING TECHNIQUES, *Judicael Ribault, Olivier Dalle* 515

Agent Directed Simulation

METAMODELLING FOR ANALYZING SCENARIOS OF URBAN CRISIS AND AREA STABILIZATION BY APPLYING INTELLIGENT AGENTS, *Agostino Bruzzone, Achille Scavotti, Marina Massei, Alberto Tremori* 521

SUPPLY CHAIN VULNERABILITY AND RESILIENCE: A STATE OF THE ART OVERVIEW, *Francesco Longo, Tuncer Ören* 527

INCORPORATING "BIG FIVE" PERSONALITY FACTORS INTO CROWD SIMULATION, *Sivakumar Jaganathan, Peter Kincaid, Thomas Clarke* 534

STUDY OF BIOLOGICALLY-INSPIRED NETWORK SYSTEMS: MAPPING COLONIES TO LARGE- 537

SCALE NETWORKS, *Ahmet Zengin, Hessam Sarjoughian, Hüseyin Ekiz*

EMERGENCE, ANTICIPATION AND MULTISIMULATION: BASES FOR CONFLICT SIMULATION, *Tuncer Ören, Francesco Longo* 546

Simulation based Optimization

PERFORMANCE EVALUATION OF QUANTUM CASCADED LASERS THROUGH VISSIM MODELING, *Mohamed Bakry El Mashade, Imbaby Ismail Mahmoud, Mohamed Said El Tokhy* 556

SELECTING THE OPTIMUM BY SEARCHING AND RANKING PROCEDURES IN SIMULATION-BASED OPTIMIZATION, *Pasquale Legato, Rina Mary Mazza* 561

MODELLING, SIMULATION AND OPTIMIZATION OF LOGISTIC SYSTEMS, *Pasquale Legato, Daniel Gullì, Roberto Trunfio* 569

RUNWAY CAPACITY OPTIMIZATION: AIRCRAFT SEQUENCING IN MIXED MODE OPERATION, *Olatunde Temitope Baruwa, Miquel Angel Piera* 579

Modeling & Simulation Methodologies, Techniques and Applications

GREAT BUT FLAWED EXPECTATIONS: ON THE IMPORTANCE OF PRESUMPTIONS AND ASTONISHMENT IN MODEL AND SIMULATION BASED RISK MANAGEMENT, *Marko Hofmann, Thomas Krieger* 586

MODELLING AND SIMULATION OF AUTONOMOUS LOGISTIC PROCESSES, *Bernd Scholz-Reiter, Torsten Hildebrandt, Jan Kolditz* 594

AN ANALYTICAL APPROACH FOR OPTIMISING THE NUMBER OF REPAIRMEN FOR LARGE SCALE, HOMOGENEOUS, MULTI-SERVER SYSTEMS, *Orhan Gemikonakli, Hadi Sanei, Enver Ever, Altan Koçyiğit* 602

A SOLUTION FOR IMPROVED MODELLING EFFICIENCY OF A MULTI-DISCIPLINARY MARINE POWER SYSTEM, *Jeroen Schuddebeurs, Patrick Norman, Ian Elders, Stuart Galloway, Campbell Booth, Graeme Burt, Judith Apsley* 610

AIRPORT TERMINALS' PERFORMANCE ANALYSIS: A CASE STUDY, *Antonio Cimino, Duilio Curcio, Giovanni Mirabelli, Enrico Papoff* 620

ON THE USE OF OPTICAL FLOW TO TEST CROWD SIMULATIONS, *David Kaup, Thomas Clarke, Linda Malone, Rex Oleson, Mario Rosa* 627

PROXEL-BASED SIMULATION OF QUEUING SYSTEMS WITH ATTRIBUTED CUSTOMERS, *Claudia Krull, Wenjing Xu, Graham Horton* 632

SIMULATION AND OPTIMIZATION OF VEHICULAR FLOWS IN A HARBOUR, *Raffaella Frattaruolo, Rosanna Manzo, Luigi Rarità* 638

A GRAPHICAL TOOL FOR THE SIMULATION OF SUPPLY CHAINS USING FLUID DYNAMIC MODELS, *Alfredo Cutolo, Carmine De Nicola, Rosanna Manzo* 647

HEURISTIC PROCEDURES FOR PROBABILISTIC PROJECT SCHEDULING, *Patrizia Beraldi, Maria Elena Bruni, Francesca Guerriero, Erika Pinto* 655

HIGHWAY TRAFFIC MODEL BASED ON CELLULAR AUTOMATA: PRELIMINARY SIMULATION RESULTS WITH CONGESTION PRICING CONSIDERATIONS, *Salvatore Di Gregorio, Renato Umeton, Andrea Bicocchi, Andrea Evangelisti, Miguel Gonzalez* 665

A FORMAL APPROACH FOR OPTIMIZED SYSTEM ENGINEERING, *Yann Pollet, Olfa Chourabi* 675

HYDROLOGICAL RISK ANALYSIS OF THE SOLIMÕES RIVER USING EXTREME VALUE THEORY, *Alexandra R. M. Almeida, Beatriz V. M. Mendes, Maria Célia S. Lopes, Gerson G. Cunha* 683

EVENT SCHEDULING MADE EASY: BASIC SIMULATION FACILITY REVISITED, *Luis Dias, Guilherme Pereira, José Oliveira* 688

Advance in Information and e-Integration in Healthcare Systems and Management

HADA: TOWARDS A GENERIC TOOL FOR DATA ANALYSIS FOR HOSPITAL SIMULATIONS, *Iván Castilla Rodríguez, Murat Gunal, Michael Pidd, Rosa Mª Aguilar Chinea* 693

A MODEL TO DESCRIBE THE HOSPITAL DRUG DISTRIBUTION SYSTEM VIA FIRST ORDER HYBRID PETRI NETS, *Mariagrazia Dotoli, Maria Pia Fanti, Agostino Marcello Mangini, Walter Ukovich* 700

COMPLEX ORGANIZATION MODELING AND SIMULATION APPROACH FOR OPERATIONAL SCENARIOS STUDY: APPLICATION TO HEALTH CARE ORGANIZATION, *Ahmed Sami REBAI, Vincent CHAPURLAT, Daniel DIEP* 706

MOBILE GAME-BASED LEARNING SIMULATIONS FOR CRISIS MANAGEMENT TRAINING. A CASE STUDY “MOGABAL FOR E-HEALTH”, *Stefano Mininel, Federica Vatta, Sara Gaion, Walter Ukovich* 712

A HIGH PERFORMANCE COMPUTING-BASED APPROACH FOR THE REALISTIC MODELING AND SIMULATION OF EEG ACTIVITY, *Federica Vatta, Stefano Mininel, Paolo Bruno, Fabio Meneghini, Francesco Di Salle* 718

Modeling & Simulation in Industrial Engineering

MODELING OF MARINE SYSTEMS AND PROCESSES AIMED AT OPTIMAL SHIP HANDLING, *Enco Tireli, Josko Dvornik, Srdan Dvornik* 724

ASSESSING POOLING POLICIES IN MULTI-RETAILER INVENTORY SYSTEM WITH LOST SALES, *Mounira Tlili, Mohamed Moalla, Jean Pierre Campagne, Zied Bahroun* 730

MODELLING AND CONTROLLER PROTOTYPING FOR UNMANNED VERTICAL TAKE OFF AND LANDING (UVTOL) VEHICLES, Alexander Martinez, Pedro Gutierrez, Claudio Rossi, Antonio Barrientos, Jaime Del Cerro, Rodrigo San Martin 738

ON THE INTEGRATED PRODUCTION AND PREVENTIVE MAINTENANCE PROBLEM IN MANUFACTURING SYSTEMS WITH BACKORDER, *Jean-Pierre Kenne, Ali Gharbi, Mounir Beit* 744

INVENTORY MANAGEMENT COSTS ANALYSIS UNDER DIFFERENT CONTROL POLICIES, *Antonio Cimino, Duilio Curcio, Giovanni Mirabelli, Enrico Papoff* 754

DEMAND FORECASTING AND LOT SIZING HEURISTICS TO GENERATE COST EFFECTIVE PRODUCTION PLANS: A SIMULATION STUDY ON A COMPANY IN THE WOOD FLOORS SECTOR, *Lorenzo Tiacci, Stefano Saetta* 762

STUDY OF VEHICULAR INTERACTION IN HETEROGENEOUS TRAFFIC FLOW ON INTERCITY HIGHWAYS USING MICROSCOPIC SIMULATION, *Venkatachalam Thamizh Arasan, Shriniwas Arkatkar* 770

GEOMETRIC AND MULTIBODY MODELING OF RIDER-MOTORCYCLE SYSTEM, *Michele Calì, Massimo Oliveri, Gaetano Sequenzia, Francesco Trovato* 780

MODELING, SIMULATION AND ERGONOMIC STANDARDS AS SUPPORT TOOLS FOR A WORKSTATION DESIGN IN MANUFACTURING SYSTEM, *Antonio Cimino, Giovanni Mirabelli* 788

OPTIMIZING TIME PERFORMANCE IN REACHABILITY TREE-BASED SIMULATION, *Miguel Mujica, Miquel Angel Piera, Mercedes Narciso* 800

FIRST PRINCIPLES MODELING OF THE LARGE HYDRON COLLIDER'S (LHC) SUPER FLUID HELIUM CRYOGENIC CIRCUIT, *Rafal Noga, Cesar de Prada* 806

SOLVING THE PALLET LOADING PROBLEM USING A COLOURED PETRI NET APPROACH, *Miquel Angel Piera, Catya Zuñiga* 811

SHOP ORDERS SCHEDULING: DISPATCHING RULES AND GENETIC ALGORITHMS BASED APPROACHES, *Antonio Cimino, Francesco Longo, Giovanni Mirabelli, Enrico Papoff* 817

SENSITIVITY ANALYSIS AND OPTIMIZATION OF DIFFERENT INVENTORY CONTROL POLICIES ALONG THE SUPPLY CHAIN, *Duilio Curcio, Francesco Longo* 824

Effective Design by Modeling & Simulation

AN APPROACH FOR FAULT DETECTION IN DEVS MODELS, *Diego M. Llarrull, Norbert Giambiasi* 831

ANALYSIS OF SHIPBOARD SURVIVABLE FIRE MAIN SYSTEMS, *Albert Ortiz, Don Dalessandro, Dong Qing, Li Bai, Saroj Biswas* 841

MODELING AND EVALUATION OF DISTINCT ALTERNATIVE DESIGNS FOR WIDE-BAND AIR-COUPLED PIEZOELECTRIC TRANSDUCERS, *Abelardo Ruíz Toledo, Tomas E. Gómez Álvarez-Arenas* 847

WORKPLACES EFFECTIVE ERGONOMIC DESIGN: A LITERATURE REVIEW, *Antonio Cimino, Duilio Curcio, Francesco Longo, Giovanni Mirabelli* 853

Author's index 863

SELF-LEARNING NAVIGATION MAPS BASED UPON DATA-DRIVEN MODELS USING RECORDED HETEROGENEOUS GPS TRACKS

Clemens Novak[a], Barbara Franz[b], Herwig Mayr[c], Michal Vesely[a]

[a]Research Center Hagenberg
Upper Austria University of Applied Sciences
Softwarepark 11, A-4232 Hagenberg, Austria

[b]Siemens PSE Austria
SIS PSE CVD CON1
Wolfgang-Pauli-Str. 2, A-4020 Linz

[c]Department of Software Engineering
Upper Austria University of Applied Sciences
Softwarepark 11, A-4232 Hagenberg, Austria

[a]{clemens.novak, michal.vesely}@fh-hagenberg.at, [b]barbara.franz@siemens.com, [c]herwig.mayr@fh-hagenberg.at

ABSTRACT

We present our innovative approach to keep navigation maps up to date by deducing map changes from recorded GPS tracks using adequate models and rules.

First, we describe, how models for receiver, mobility and terrain can be generated from adequately preprocessed recorded GPS tracks. These models are used by a server in order to predict plausible extensions of available navigation maps. In order to allow for multimodal track sources (pedestrians, automobilists, bicyclists, horseback riders, etc.), geometrical matches have to be further checked for plausibility. We give examples of such plausibility rules we have developed for this purpose.

The main benefits of our development are better maps and better guidance for various classes of possible users, from pedestrian, over cross-country skier, to bus driver, to name just a few.

Keywords: Global Positioning System – GPS, digital navigation maps, data-driven models, incremental map enhancement

1. MOTIVATION: SELF-LEARNING NAVIGATION MAPS

Although the navigation hardware market has changed rapidly during the last few years, the digital navigation maps market has remained rather inflexible: In some regions of the world, digital maps are still rudimental or even not available at all.

For instance, the off-road personal navigation for pedestrians as well as for outdoor and leisure activities (like mountain biking, hiking, horseback riding, or track skiing) is just in its infancy, and it is based on point-to-point navigation as opposed to full-fledged navigable vector maps of common vehicle navigation systems.

Thus, flexible, adaptive navigation using data that can be generated, updated, and personalized individually as well as for the public benefit still remains an interesting research topic with good chances for commercial value in the near future.

This contribution describes our approach to incremental navigation map generation based on data-driven models generated from recorded heterogeneous GPS tracks (Hofmann-Wellenhof, B., Lichtenegger, H., and Collins, J., 2001) gathered through tracing the paths of entities with heterogeneous mobility models (pedestrians, bicyclists, cars, etc.).

2. PREPROCESSING NOISY DATA

In order to enable adequate processing of the track data, i.e. filter-based preprocessing and verification, as well as the merging of the data with the existing map, we base our threshold values for algorithms on systematic data-driven models.

The data preprocessing covers filtering incorrect and insignificant track data. Since tracks based solely on the GPS data tend to be inaccurate in certain situations (e.g., reflections in built-up areas, changing satellite trajectories; cf. (Hofmann-Wellenhof, B., Lichtenegger, H., and Collins, J., 2001)), a track filter was implemented to filter out objectionable positioning errors: *Outliers* are removed using topological considerations and mobility constraints (velocity, acceleration) of the recording entity. *Accumulations* – which may for instance occur due to the recording entity staying at the same location for some time – are removed by appropriate clustering of GPS points and removing superfluous points.

We base our filter algorithms on configurable thresholds defined by the models as described in the following chapter. The composed data is processed by a proprietary graph matching algorithm combined with a set of models that take the varieties of input data into consideration. This approach provides the core functionality of incremental map learning.

For details on preprocessing and graph matching approach, we refer to (Vesely, M., Novak, C., Reh, A., and Mayr, H., 2008).

3. INDUCTIVELY GENERATED MODELS

From the preprocessed data, suitable models have to be derived describing the entity and its mobility as well as GPS signal quality depending on the terrain and the receiving device. These models can then be used for assessing the quality and topological soundness of future gathered data.

Depending on the type of entity used for recording the route (vehicle, pedestrian, bicyclist, etc.), a *mobility model* can be defined describing maximum velocity, probable and maximum curve radii depending on the velocity, among other parameters.

According to this model, and additionally taking the respective values from the GPS track data into account, the most probable path within the limits of the track data can be determined.

If the same geographic path is recorded several times by the same entity or different ones, the most probable course of the real path or road can be determined by a suitable weighted averaging method which uses additional models – like Kalman filters – as detailed in (Mayr, 2007).

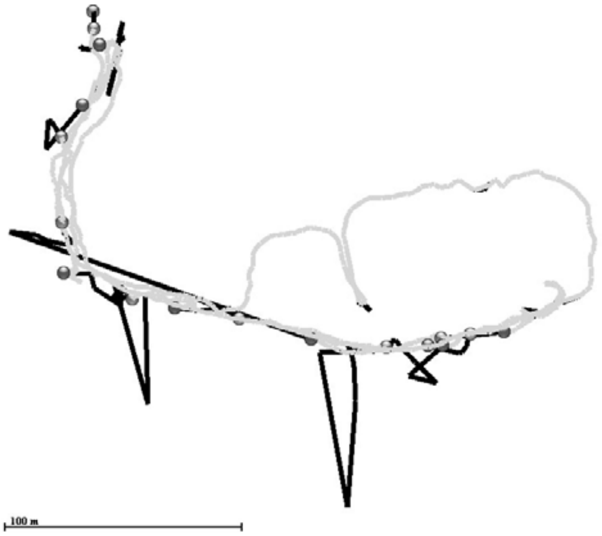


Figure 1: Filtered route (gray) on original (black)

Additionally to or instead of the Kalman filter used for preprocessing, a recorded route can be preprocessed by checking its attributes for plausibility (cf. Fig. 1). Therefore we define attributes, e.g. height and velocity of each recorded point as well as at the distance to the other points that confirm whether a recorded point is a plausible point that represents a real part of the recorded path. If the position data is incorrect because of errors due to inaccuracies in the position measurement or receiver-specific faults, it has to be filtered out.

For the preprocessing with plausibility checks three models are needed:

3.1. Receiver model

The quality of GPS, besides its standard error deviation due to atmospheric effects and clock synchronization, is heavily dependent on the satellite geometry. Every change in the satellite constellation can cause outliers (see (Hofmann-Wellenhof, B., Lichtenegger, H., and Collins, J., 2001) for details). Additionally, slow movement (< 4 km/h) causes GPS-positions to drift, thus deteriorating the quality of the recorded track. In some areas signal masking occurs and causes either signal loss or deficient positioning which makes the data inadequate for our self-learning system.

The quality of GPS data can be measured by *delusion of precision* (*DOP*) – the higher the DOP value the lower the quality. The receiver model defines thresholds for horizontal, vertical and positional DOPs (*HDOP*, *PDOP*, *VDOP*). Recorded positions that show DOPs above these thresholds don't suffice specified quality and are filtered out.

3.2. Mobility model

The mobility model entails attributes that describe the type of entity used for recording the route. These typical characteristics can be used for plausibility checks of the position data by comparing the recorded data with the model parameters.

Each entity has a maximum velocity that can be defined in the model. For pedestrians the absolute maximum velocity is assumed as 20 km/h, for cars as 260 km/h, etc., as points with a velocity above these defined thresholds are considered not plausible and thus cannot be used for the enhancement of maps.

The same applies to the difference in altitude per seconds as well as possible heading changes and curve radii depending on velocity. If the recorded curve radius is smaller than the minimum radius that would be possible at the recorded speed, the track data is not accurate at this part of the route. Table 1 (adapted from (FGSV, 1995)) gives an overview of minimum curve radii (r_{min}) depending on velocity (v) for the mobility model "car".

v [km/h]	r_{min} [m]
< 50	80
51 – 60	120
61 – 70	180
71 – 80	250
81 – 90	340
91 – 100	450
101 – 120	720
> 121	n.a.

Table 1: Minimum curve radii depending on velocity

3.3. Terrain model

After filtering a series of points it can be necessary to divide the preprocessed route into two or more new routes to enhance the learning process. Position values that are missing due to filtering or signal masking may lead to implicated connections between points that are not directly connected in reality. Therefore the system may learn incorrect data, because curves or intersections are not recorded.

Thus, to avoid negative influence on the enhancement of the map, the terrain model defines a distance threshold that specifies the maximum plausible distance between two points. Our empirical study shows that for instance for pedestrians the distance threshold should be set between 10-15 meters, if there are no mapped tunnels near the recorded route. Pedestrian routes can be very short which is why intersections and even whole parts of a route can be missed easily.

The threshold not only depends on the terrain of the route (e.g. existing tunnels) and on the type of entity, but also on the average point distance due to the velocity of the entity and the sample rate.

4. SYSTEM ARCHITECTURE

In order to enable the capability of extending and personalizing digital navigation data and, thus, continuously improve their quality and actuality, we have developed a software system that allows acquiring GPS tracks and environment-based attribute data automatically from the recorded GPS tracks, as well as integrating these into existing data (Fig. 2).

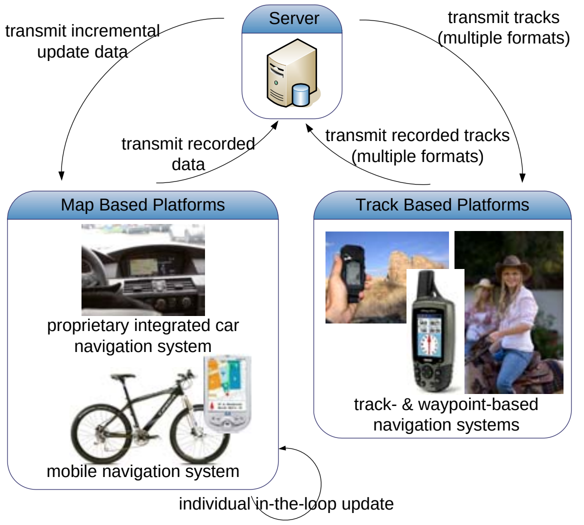

Server
transmit incremental update data
transmit tracks (multiple formats)
transmit recorded data
transmit recorded tracks (multiple formats)
Map Based Platforms
proprietary integrated car navigation system
mobile navigation system
Track Based Platforms
track- & waypoint-based navigation systems
individual in-the-loop update

Figure 2: System Architecture Overview

Our system is comprised of four essential components, 1. navigation entities, 2. central server, 3. data exchange interfaces, and 4. a data management component, which are described in more detail below.

4.1. Navigation Entities

Basically, all kinds of GPS-enabled mobile devices can be used for recording GPS tracks to be uploaded to the server, if their output data format is among the supported ones.

In order to prove the feasibility of our approach as a whole on concrete implementations, we utilize two map-based navigation systems for mobile devices: an adapted car navigation system and a PDA-based mobile navigation and recording system developed from scratch. For more details on the navigation entities, we refer to (Vesely, M., Novak, C., Reh, A., and Mayr, H., 2008).

4.2. Central Server

The server comprises the ability to receive the data from the recording entities, process them, as well as generate and administer incremental map updates to be offered via a public service. This results in the following module design (cf. Fig. 3):

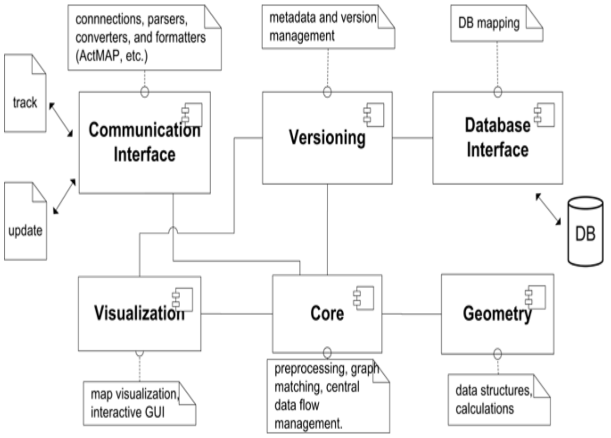

connnections, parsers, converters, and formatters (ActMAP, etc.)
metadata and version management
DB mapping
track
update
Communication Interface
Versioning
Database Interface
DB
Visualization
Core
Geometry
map visualization, interactive GUI
preprocessing, graph matching, central data flow management.
data structures, calculations

Figure 3: Server Modules Overview

The *communication interface module* administers multiple connection interfaces for receiving the data. The parsers and converters for the most common track data formats are consolidated in a separate *formatters sub module*.

After extracting the raw track data to internal data structures pooled in the *geometry module*, data processing is done in the *core module*.

In order to enable manual inspection of our approaches and algorithms, we developed a *visualization module* that enables visual evaluation, editing, and further management of the digital map data stored in the relational database.

For the proper management of a vast quantity of incremental map updates, we have developed a metadata-based *versioning module*.

All of the above modules rely on the central *database interface module* which provides an object-oriented abstraction layer above an exchangeable relational database, allowing manageable parceling and adaptation of the map data.

For details on our system architecture as well as on our algorithmic solutions regarding the incremental map generation and graph matching technique, we refer to (Vesely and Mayr, 2007; Vesely, M., Novak, C., Reh, A., and Mayr, H., 2008).

4.3. Data Exchange Interfaces

In order to enable the exchange of attributes and other metadata related to the tracks, we utilize the XML-based GPX format (GPS Exchange Format, 2008) as the main data exchange format. Thereby the users can either upload their data directly from their navigation system (if supported, like in our PDA-system), or upload the raw output data via a web portal and specify the metadata and attributes using an interactive web interface.

We base our incremental map data update strategy on the specification of the EU-supported project "ActMAP", which specifies the strategies and the exchange format for online incremental updating of digital map databases. The exchange format specification is based on the ISO standard *Geographic Data Files* (GDF). For further details on ActMAP, we refer to (Otto, H., Beuk, L., Aleksic, M., Meier, J., Löwenau, J., Flament, M., Guarise, A., Bracht, A., Capra, L., Bruns, K., and Sabel, H., 2004).

4.4. Data Management

The navigation map is based on a graph of vertices and edges, where the vertices represent the junctions of more than two edges. In order to provide a serviceable map, further edge attribute data must be contained in the map, e.g. the shape points which approximate the true course of the edges, categorization data, direction of trafficability, average time of travel on an edge, etc. The data are stored in a relational database.

In order to enforce performance and minimize memory load of algorithms working with huge amounts of graph data, the earth surface is logically divided into tiles, so-called *parcels*. They are of the same geographical size and their indexing is based on geo-coordinates. The indexes of neighboring parcels can be identified by simple calculations, thereby allowing for fast and systematic retrieval of relevant data.

For the appropriate management of all generated map changes in the relational database, all elements in the database are version-controlled in order to allow for backward compatibility, version-switching and update definition generation for a specific map revision at any time.

Since our PDA-based navigation solution has to provide the same data update functionality independent from the server ("in-the-loop"), we utilize the server's data management concept in our mobile solution as well.

5. USING OUR MODEL BASE FOR PREDICTION OF PLAUSIBLE EXTENSIONS OF AVAILABLE NAVIGATION MAPS

Since our graph matching algorithm (Vesely, M., Novak, C., Reh, A., and Mayr, H., 2008) relies on the simple map matching approach, its results reflect only the pure geometrical match of the given GPS track to the available map data. In order to improve the decision of the graph matcher, we use a model-driven approach to check the plausibility of the geometrical match.

A typical example for these induced plausibility checks is a combination of a pedestrian-recorded track misleadingly matched by the graph matcher to an edge classified as highway. Based on the according mobility model of the recorded track the graph matcher decides whether a new edge should be generated, or the affected section of the track is matched to the corresponding edge in the available map. Fig. 2 shows that, using the rule base, our algorithm correctly implies that no intersection between the pedestrian route and the highway exists (black – no match, dark gray – match, light gray – highway).

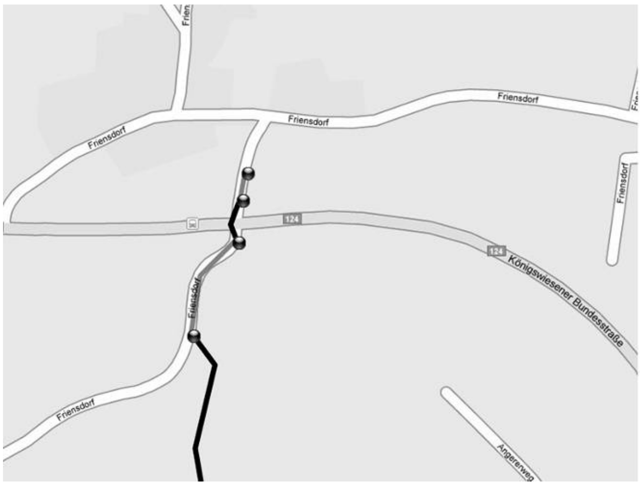


Figure 4: Pedestrian track crossing a highway

Another example may be a new highway bridge built over an existing road. Geometrically, such a bridge may be misleadingly classified as a crossroads, since in current map data of navigation systems altitude information is not stored for navigation graph elements. However, the correct inclusion of the bridge into the existing map data may be deduced from the driving behavior of automobilists passing the bridge. If several recorded tracks show that cars pass the "crossroads" without reducing the speed, our rule base correctly implies that no intersection between the two roads do exist and the map data is correctly adapted.

Some routes restrict which type of road users are allowed to access. If the user who recorded the track is not allowed to use the mapped route, it is not plausible that this route is the way the user walked or drove on. A car route for example can never be matched to a pedestrian way, a route recorded by a mountain biker never to a freeway, etc.

The same applies to a driving restriction for certain road users or at certain times. For example a truck may be transporting fuel during recording a track. If this track goes past a road leading through a water protection area, the graph matcher would match the route to this road. According to our rule base there exists a cargo restriction rule whereas the mapped road is not considered a plausible representation of the driven track.

These examples show that some rules are considered expedient only for certain mobility models. Another example is the check if the gross vehicle weight exceeds the permissible maximum weight of the matched road segment. In this case the entity cannot be driven there (legally). This rule, as well as the same rule applied to height or width doesn't make sense for pedestrians or bicyclists.

For checking gathered geometric constellations are compared to our defined models by our rule base which concludes whether a match is plausible in order that the map data can be adapted correctly. If a map or a recorded path contains new or additional information, our rule base as well as our models can be easily extended.

6. A STEP-BY-STEP APPROACH TO PLAUSIBILITY CHECKS

To check the plausibility of the geometrical match we define a rule base that classifies whether a match defined by the graph matcher is a plausible one, using three models. Therefore the preprocessing models are extended to combine information about the map data as well as additional rule specific information:

- *Terrain model for map data:* Digital maps do not only contain information about the position of streets and their form but also additional characteristics that can be used for our rule base. The Navteq Core Map for instance includes 14 different attribute categories that comprise 204 map attributes which are used for the map representation as well as for route planning (Navteq, 2008).
  Our terrain model for map data combines street information like intersections, street type or direction as well as restrictions concerning users, velocity, direction, turns , etc.

- *Terrain model for route data:* GPS receivers do not only record the position but also information about the date, quality, velocity, etc., which are combined in a terrain model for route data to be used in our rule base.

- *Mobility model:* Additionally to the parameters needed for the preprocessing steps, we define the height, width and gross weight of the entity as well as cargo for checking restrictions and regulations.

According to our models we implement a rule base that checks step-by-step whether a recorded track or parts of it are represented in a plausible way by the results of the graph matcher. The match is considered plausible if no rule applies that rates the match as not plausible. If there is more than one match rated plausible, the one with the total best result in the graph matching and the plausibility check is considered the best match for the recorded track.

The rules defined in our rule base are applied to the terrain model and the motion model of the currently observed track point as well as to the terrain model of the point on the edge that is considered a geometrically plausible match by the graph matcher. If the data is comprised by different entities and therefore more than one mobility model is used, some rules have to be checked several times.

In the following, examples of our rules are described, abbreviating the terrain model for map data as *TM*, the terrain model for route data as *TR*, the mobility model as *M*, “plausible” as *p* and “not plausible” as *u*:

- $\neg EQ((TM), type(M)) \rightarrow u$

  On some streets there are only certain groups of users allowed. If the type of the entity, defined in the mobility model, is not declared as an admissible type of the map edge (*EQ*), the map edge candidate is not considered to represent the recorded route. If there is no information in the terrain model of the map about which users are allowed, the comparison is made using the type of the map street. Therefore the rule base needs the explicit knowledge that in pedestrian zones only pedestrians are allowed, on freeways only cars, etc. This information may differ from country to country, wherefore in certain cases also the country is needed.

- $v(TR) + \varepsilon > v_{max}(TM) \rightarrow u$

  If the driven velocity exceeds the velocity that is allowed on the matched edge by more than the threshold ε, the match is not plausible.

- $v_{max}(B) < v_{min}(TM) \rightarrow u$

  If the maximum possible velocity of the entity, defined in the mobility model, is lower than the minimum allowed velocity on the matched map path, then the edge candidate can neither possibly represent the recorded path nor cross it.

- $H(TR) \pm \varepsilon \neq H(TM) \rightarrow u$

  If the sea level is defined in the map and exceeds the sea level recorded at the current point by a predefined threshold ε either there is no representation of the recorded path in the map or another candidate, that was considered geometrically plausible by the graph matcher, has to be checked for logical plausibility.

- $one\text{-}way(TM)\ \&\ (dir(TM) \neq dir(TR)) \rightarrow u$

  If the matched street is a one way street and the direction (*dir*) of the street does not match the direction of the recorded route, the match is not considered plausible.

- $restriction(TM, type(M), cargo(M), date(TR)) \rightarrow u$

  If there is a driving restriction on the matched street at the date and time when the path was recorded for the type of mobility model and/or regulations concerning the transport of certain goods, the street is no candidate for representing or crossing the recorded path.

- $EQ(location(TM), DOP(TR)) \rightarrow p$

At some locations, like for instance in forests or street canyons, the quality of positioning with GPS is often low. The plausibility of a match increases if the rule applies that the location of the matched route fits the recorded DOP (*EQ*).

- $\neg EQ(intersection(TM), v_{max}(TM), type(M)) \rightarrow u$

  If the recorded route intersects a street on the map, our rule base checks whether an intersection is valid (*EQ*).

The more detailed the models are, the more rules can be concluded to reason about the plausibility of the graph matching result and therefore enhance the quality of the deduction process. For further details on our rule base we refer to (Franz, 2008).

7. BENEFITS FROM OUR APPROACH

Our model-based approach allows the integration of various types of recorded data into one basic set and enables the computation of the most probable map graph representing real world paths and roads used by the recording entities. This results in the following major improvements:

- *Better Maps*: By individual additions of missing map segments as well as broadcasting them via a server (after verification), the continuous incremental update of navigation maps ("improving and extending the map with each usage") based on data integration from different sources (pedestrians, bicyclists, automobilists, etc.) results in harmonized, accurate, and up-to-date maps, particularly for highly frequented areas.

- *Better Guidance*: Our approach enables the creation of personalized maps adjusted to the personal preferences and needs of the user, e.g. training courses for sport and leisure activities (e.g., scenic and smooth routes, etc.), footpath-maps for pedestrians, specialized maps suited to navigation of people with disabilities like wheel-chair users, visually-challenged or elderly people, etc.

Moreover, due to the modular architecture of our system, it can be tailored to the target group's needs, thus enabling future navigation systems to utilize our modules selectively, according to requirements.

ACKNOWLEDGMENTS

Parts of this work have been financed by the Austrian government (FFG), grant no. 811 406 (FH*plus* program). Other financially contributing partners to this project are Siemens Austria and Intersport Austria.

REFERENCES

Hofmann-Wellenhof, B., Lichtenegger, H., and Collins, J., 2001. *GPS: Theory and Practice*. Wien New York : Springer-Verlag.

Vesely, M., and Mayr, H., 2007. Using Self-adapting Navigation Data for Intelligent, Personalized Vehicle Guidance. *In: Proc. R. Moreno-Díaz et al. (Eds.): EUROCAST 2007, LNCS 4739*, pp. 1097–1104. Feb. 12-16, Las Palmas de G.C., Spain.

Vesely, M., Novak, C., Reh, A., and Mayr, H., 2008. Incremental Navigation Map Enhancement with GPS Tracks from Heterogeneous Sources. *In Proc. 2008 International Conference on Machine Learning Models, Technologies and Application*, pp. 787-793. July 14-17, Las Vegas, Nevada, USA.

Mayr, H., 2006. Model-Based Navigation Using GPS: One Step Closer to Intelligent, Incremental Maps. *In Proc. International Mediterranean Modeling Multiconference*, pp. 641-646, Oct. 4-6, Barcelona, Spain.

Mayr, H., 2007. I-Navigate: Intelligent, Self-adapting Navigation Maps. *In Proc. 14th IEEE Intl. Conference and Workshop on the Engineering of Computer Based Systems,* pp. 397–402, March 26-29, Tucson, Arizona, USA.

FGSV, 1995. *Richtlinien für die Anlage von Straßen, Teil: Linienführung RAS-L (Guidelines for the Creation of Roads, Part: Trajectory Recommendations RAS-L; in German)*. Forschungsgesellschaft für Straßen- und Verkehrswesen, Arbeitsgruppe Straßenentwurf, Düsseldorf (Germany).

Franz, B., 2008. *Plausibilitätsprüfung von Karten für Navigationssysteme auf Basis aufgezeichneter Strecken (Plausibility Checking of Navigation Maps Based Upon Recorded Tracks; in German)*. Master's thesis, School of Informatics, Communication and Media, Upper Austria University of Applied Sciences, Hagenberg, Austria.

NAVTEQ, 2008. *NAVTEQ Core Map*, Available from: http://developer.nav teq.com.

GPS Exchange Format, 2008. *GPX: The GPS Exchange Format*, Available from: http://www.topgrafix.com/gpx.asp.

Otto, H., Beuk, L., Aleksic, M., Meier, J., Löwenau, J., Flament, M., Guarise, A., Bracht, A., Capra, L., Bruns, K., and Sabel, H., 2004. *ActMAP ISO Input.* Technical Report D 3.3, ERTICO.

BIOMIMETIC MIDDLEWARE FOR WIRELESS SENSOR NETWORKS

Zenon Chaczko

ICT Group,
Faculty of Engineering, University of Technology, Sydney, Australia

zenon.chaczko@uts.edu.au

ABSTRACT

An agent based, Biomimetic (biology inspired) Middleware for resource constrained systems such as Wireless Sensor Networks (WSNs) by emulating nature is able to provide infrastructure oriented services that are characterised by such autonomic and autopoietic properties such as: (1) self-organisation, (2) self–shaping, (3) self-monitoring and self-healing. The paper aims at explaining how these fundamental properties if imprinted on executing agents can help in construction of reliable, cooperative and sustainable information ecosystems such as biomimetic middleware. This can occur through application of genetic evolution and immuno-computing paradigms (i.e. selection inhibition, random enabling/inhibiting, preferential attachment, birth, growth and death) .

Keywords: middleware design, autopoietics, biomimetics, resource constrained system

1. INTRODUCTION

Recent developments in software engineering and well sensor networks technology, biological science as advances in the complex systems theory have brought a new perspective on the different aspects of modeling (Bruzzone and Longo 2005), (Effroni et al. 2005), simulation and design of resource constrained systems. Surprisingly, in a relatively short time very strong links have been established between these disciplines resulting in proliferation of various biology inspired (biomimetic) computing models, methodologies and techniques (Keele and Wray 2005), (Kennedy 2006), (Krishnamurthy et al. 2006a), (Krishnamurthy and Murthy 2006b).

Although fundamentals of biomimetics have been adopted in many areas of engineering, the theoretical principles, concept, and characteristics of this new research domain have not yet been well established or clearly defined. Application of biomimetics in design of software systems is not an exception here (Ishida 1997), (Bonabeau et al. 1999), (Dressler and Krüger, 2004). This is can be mainly attributed to interdisciplinary nature of the field. The main thrust of our research into models of BIomimetic Middleware (BIM) systems is to explore how interdisciplinary insights could integrate systems theory, biology and software engineering in order to achieve a beneficial impact on methodology, architecture and engineering of newer generation of infrastructure oriented and sustainable software for Wireless Sensor Networks (Gustavsson and Fredriksson 2003), (Hull 2004).

A traditional approach for modeling and designing software intensive systems puts a strong focus on defining required resources for construction, testing and then operation of the implemented system. Far away from proper engineering practices the traditional approaches are additionally limited by economics, incompleteness of specifications, imprecise system performance information, over-provisioning as well as other factors. In the proposed biomimetic approach, beyond of the analytical concerns for architecture and design of software infrastructure we provide a consideration to the autonomic and autopoietic properties that may be used to improve the resource management of network systems. Biomimetic approaches provide an interesting modelling framework within which we could study the various interactions among the components of a software system, and how these interactions can influence the function and behaviour of the system under consideration. Architectural models and methods of resource and congestion management for dynamic wireless sensor networks (Wan et al. 2003), (Hull et al. 2004), (Wang et al. 2005) are required to synergise subordination, tolerance and conflict (collision) relations among its components (Chaczko, Klempous, Nikodem and Nikodem, 2007). In particular, in our research we study how the biomimetic can improve the architectural qualities of software infrastructure.

In this paper we aim to demonstrate that the structure of WSNs that adheres biomimetic principles can improve aspects of system cooperativity and robustness. In our opinion, the quality of system robustness can be viewed as the system's ability to recognise the problem (self-monitoring) and to know what to do in order to repair the system itself (self-repair). The prime objective is to demonstrate that we could enhance modeling and designing of the software systems (and middleware in particular) using the biomimetic rules and autopoietic (wher autopoietic = *auto – αυτό* in Greek and for self- and *poiesis – ποίησις* in Greek for creation or production) principles. Before we could claim a major methodological breakthrough we have to demonstrate that by applying the biomimetic approach we can obtain distribution and provisioning of resources (in constrained environment) as good (or better) as bio-systems are able to manage. However, it still remains a question if we actually can verify their optimality. Within our work we strive at proving that we are actually able to do it and derive some primitive measure of its optimality. It will be our future work to verify wider aspects of the proposition.

2. AUTOPOIETICS AND AUTONOMICS

In the autopoietic theory originally proposed by the H. Maturana and F. Varela in 1980 - a living system (biological organism) can be defined as a circular, autocatalytic-like and survival oriented process (Maturana and Varela 1980), (Maturana and Varela, 1987), (Livingston 2006). The theory emphasises aspects of closure of living systems (bio-organisms) thus can be perceived as an adequate alternative for the excessive attention paid to aspects of openness in theory of open systems. The self-* phenomena (where self-* denotes various autonomic attributes such as: self-organisation, self-healing, self-adaptation, self-monitoring, etc.) can be perceived as autopoietic phenomena. Software systems equipped with autonomic properties are capable of performing and managing majority of their operations completely by themselves or only with minimal level of human intervention. Autonomic network systems including autonomic wireless networks by adopting the concepts of autonomics are capable of self-managing all its elements and data communication links. In autonomic scenarios, wireless communication is complex and requires form designers to consider a number of problems related to sensor localisation, clustering, routing, energy management as well as various constraint conditions related to transmission collisions, multipath interference, obstructions that adversely impact the data throughput of high bandwidth communications robustness, reliability and scalability (Loureiro and Ruiz 2007). In order to face the challenges related to implementation of autonomic functions in WSN we propose a model of biomimetic software infrastructure (middleware) system that is characterised by the following fundamental properties:

2.1. Self-organisation

The phenomenon of self organisation is pervasive in natural systems (Eigen and Schuster 1982), (Kauffman 1993), (Eigen and Winkler 1993), (Kauffman 1995), (Bak 1996). It can be defined as a tendency of systems to evolve into a more organised form in the absence of external influence (or pressure) as a response to the system's environment; It can be also perceived as a collective, cooperative and coordinated process in which system's components reach a higher level of organisation while interacting among each other and with the environment. BIM software systems incorporating a colony (ecology) of numerous of components (agents) communicating, interacting (cooperating and competing) among each other and with the environment. Collectively and cooperatively these components can perform various tasks as a team, coordinating their activities to obtain an optimal efficiency. The sum of dynamic behaviour arising due to cooperative (competitive in resource constrained situations) interactions between different parts of the BIM could result in a coherent behaviour of the system as a whole. The system sensitivity relies on capability of middleware services to execute major changes (large amount of elements can be affected) even if the value of an observed (control) parameter is modified by a small volume only. These changes bare resemblance to the phenomenon of a phase shift in the domain of physics, where a local event can result in a global transition (a spontaneous "jump" to state(s) of much greater organisational complexity (Prigogine et al. 1993) in which new properties may dramatically re-emerge. Attributes of the BIM can't be anticipated in advance from the properties of the individual agent interactions. Following the phenomenon of emergence, various observed degrees of freedom originating from a network's parts collapse into a reduced amount of newer components, now with a smaller number of adjustable parameters that are relevant at a global level. Since by principle, the self-organisation properties cannot be predetermined the wireless sensor network system facilitated by middleware evolves to a new configuration or to a qualitatively different level that is compliant with the global system and environmental constraints. The robustness of the self-organising system can be then measured by a rate at which the system in its newer configuration is able to detect and handle its faults.

2.2. Self–shaping

We can define self-shaping it as allometric as scale-invariant, power-law scaling (Mitzenmacher 2003), (Newman 2005), (Clauset 2007), (Grant 2007) property of a system that addresses combined aspects of self-adaptation and self-optimisation. It may be understood as capability of the system to alter or adjust its structure, size and constraints (rates) of metabolic processes according to its varying internal and external surrounding conditions (Bunk 1998), (West & Brown 2005), (West et al. 2007). Shape alterations follow the optimal choice principles by adjusting various parameters in correspondence to the system behaviour (self-optimisation); In resource constrained networks there is a requirement for software infrastructure to be able to self-modify the network shape, adapt to variable levels of available resources and changes in the environment. To promote the system emergent properties such as robustness, survivability or resilience of the BIM, this needs to occur preserving aspects of scale-invariant relations (Dorogovtsev and Mendes 2003) and ego-morphic rules (Chaczko and Resconi 2007). In their work Dorogovtsev and Mendes (2003) argue that this property is fundamental as it ensures presence of all other autopoietic properties (i.e. self-organisation) in the network system.

2.3. Self-adaptation

The self-adaptation property can be seen as a capability of a system to modify/alter or adjust its internal structure or/and behaviour according to varying conditions in its surrounding.

2.4. Self-healing

The self-healing is to be perceived as a capability that allows to automatically detect, localise, diagnose and heal (repair) failures (faults and errors); The process of self-healing is adaptive, fault-tolerant and inter-dependent with the mechanism of self-monitoring.

2.5. Robustness

Robustness is also known as the fault tolerance can be seen as a capability of the system to resist or tolerate noise, disturbances, faults, stress, modification in system architecture (structure and behaviour) or changes in external ecosystem without causing negative effects on system's functions and without having long lasting effects on its structural composition and dynamics;

2.6. Cooperativeness

Characteristics of cooperativeness is to perceived as a system's capability to stimulate collective and cooperative interactions among its components. Components can perform various tasks in teams, coordinating their activities to obtain an optimal efficiency. In reality, there are various degrees of cooperation/competitive (in resource constrained situations) behaviour at place. The sum of dynamic behaviour arising from cooperative interactions between different parts of the system could decide about coherent behaviour and robustness of the whole system. Thus robustness of a system can be measured as a rate at which its components (resources) are repaired or replaced against the rate ($\mathbf{R}_s = dC_r/dC_f$) they disappear.

2.7. Resilience

The property of resilience can seen as capability to absorb and even utilise (frequently with advantageous results) noise, disturbances and changes that attain them, in order to sustain and persist without any qualitative changes in the architecture of the system.

BIM for resource constrained systems need to be flexible to change and resistant to damage or faults; the systems should be able to self-modify their past behaviour and adapt to newly allocated tasks, changes in levels of available resources or changes in environment. BIMs based WSN are robust as they are able to tolerate failures, non-cooperative behaviour or conflict (collision) relation among its components. We should be able to include software functions that apply genetic mutation and reproduction to seed autonomic and autopoietic properties in WSN.

3. ALLOMETRY AND SOFTWARE DESIGN

In order for BIM software system infrastructure to support management of the WSN according to varying levels of available resources, tasks and changes in the environment we need to model, design and then implement the self-shaping (i.e. self modification of the network topology) function requirement. It is suggested that this new type of the system property can be ensured if we follow allometric laws that are often pertinent to living systems (McMahon and Bonner 1983), (Calder 1984), (Niklas 1994), (Darveau 2002), Gillooly and Allen 2007). Typical examples of common allometric laws are:

- **Murray's law** can be observed in sections of human and animal vascular systems (LaBarbera 1990). To a degree it is also satisfied in plants. Durand (2006) argues that application of Murray's law could be very useful in performance analysis and optimisation of networked systems such as transportation infrastructure or electric power grids. In recent times, the principles of optimisation analysis and resource management of WSNs has also been a topic of an intense debate (Das et al. 2004, Hull et al. 2004.). In our work we aim to demonstrate that, we can better characterise the structure of sensor networks that can with some limitations satisfy the global optimisation of communication networks by applying the Murray's law principle. We take into account the Euclidean geometry agreeing that the optimisation has to be achieved within constraints of the classic geometry. Originally Murray's law referred to the radii (R_i) of mammalian blood vessels, since then many researchers have indicated that the exponent λ is ~3 in the following equation:

  $$R_k^\lambda - R_{k+1,1}^\lambda + R_{k+1,2}^\lambda + \cdots + R_{k+1,\eta}^\lambda, \qquad (1)$$

  A more general form of Murray's law is demonstrated in Fig. 1.

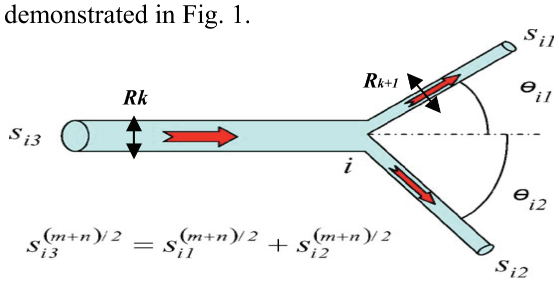
  

  Figure 1: The Relation Between Cross-Sectional Areas of Adjoining Channels in Minimal Resistance Networks (Durand 2006).

- **Kleiber's law** (Kleiber 1932), although somewhat controversial the law postulates the proportionality between organismic metabolic rate q_0 and body mass M raised to the power 3/4 defined as follows:

  $$q_o \sim M^{3/4}, \qquad (2)$$

He and Chen (2003) estimated that the power exponent is to be set respectively 1/2, 2/3, 3/4 for 1- dimensional, 2-dimensional and 3-dimensional organisms thus reconciling the 3/4 (Kleiber's law) and the 2/3 (Rubner law) values for power exponent in the lawand generalising the law which can then be defined as:

$$qo \sim M^{d/d+1}, \qquad (3)$$

where d is the dimensionality, qo the metabolic rate and M is the mass of the organism (He & Chen, 2003). Drawing from the fractal cell geometry concepts He and Zhang (2004) brought about a controversial idea that the fractal dimension analysis of the exponents leads to a 4/5 allometric scaling law for the human brain thus concluding a 5th dimension of life. Application of Kleiber's law finds many followers and opponents (Bejan, 2002), (Kozlowski et al. 2003), (Kozlowski and Konarzewski, 2004), (Makarieva at al. 2005), (Chaui-Berlinck, 2006), (West and Brown 2005), (Brown, West and Enquist 2005), (Phillips 2006), (West et al. 2007), (Enquist et al. 2007) of the metabolic theory of ecology which predicts that there is a strong relationship between metabolic rates and body size, mass and even temperature in all biological organisms; beginning from unicellular bacteria and finishing on plants and animals. This occurs over 27 orders of magnitude, based on the laws of geometry and physics of networks.

- **The Square-Cube law** is commonly applied in engineering design. Drawn from the mathematics of proportion is a principle that states that resources consumed can be measured in 3 or more different dimensions (and regarded as a volume). First demonstrated in 1638 by Galileo's *Two New Sciences*. It states that: "*When an object undergoes a proportional increase in size, its new volume is proportional to the cube of the multiplier and its new surface area is proportional to the square of the multiplie"*. This can be expressed as:

  $$v_2 = v_1 \left(l_2/l_1\right)^3, \qquad (4)$$

  where v_1 is the original volume, v_2 is the new volume, ℓ_1 is the original length and ℓ_2 is the new length. It is worth to note that it doesn't matter which length is actually used.

  $$A_2 = A_1 \left(l_2/l_1\right)^3, \qquad (5)$$

  where A_1 is the original surface area and A_2 is the new surface area. For example, if a cube solid with a side length of 1m were doubled its size, its volume would be 8 m^3 and its surface area would be 24 m^2. This principle applies to geometry of all solids. The law would be relatively easily enforced if applied in 3D topology of networks or translated into other WSN network measures.

- The proportionality between breathing and heart beating times t and body mass M raised to the power 1/4:

  $$t \sim M^{1/4}, \qquad (6)$$

- Mass transfer contact area A and body mass M:

  $$A \sim M^{7/8}, \qquad (7)$$

- The proportionality between the optimal cruising speed V_{opt} of flying bodies (insects, birds, airplanes) and body mass M in *kg* raised to the power 1 / 6:

  $$V_{opt} \sim 30.M^{1/6} \text{m.s}^{-1}, \qquad (8)$$

4. CONSTRUCTAL THEORY AND LIMITS OF SELF-SHAPING

We are aware that biomimetic approach has serious limitations. Stringently applied biological and allometric rules may also pose threats to the design and construction of software infrastructure systems. It is a fact that millions of years of biology have proven to be very successful. However, there are good reasons to believe that abrogating these laws for artificially (human) created systems would likewise cause sadden system instabilities and even a complete (unrecoverable) failure (death). After all as a rule biological systems die.

In constructal theory defined in 1996 (Bejan 2000, 2002, 2006), (Reis et al. 2004) its author Adrian Bejan postulated that: *"For a finite-size system to persist in time (to live), it must evolve in such a way that it provides easier access to the imposed currents that flow through it ..."*. In its essence, the constructal principle states that every system is destined to remain imperfect, however the constructal principle generates the perfect form, here understood as the least imperfect all forms possible. According to the theory, the best that we could do when designing and implementing software based systems is to optimally distribute the imperfections among components of the system. Presumably such optimal distribution of imperfection among components of the system will determine the final shape or topology of the distributed system under study. The limits of self-shaping can arise from agent/node traffic trajectories and a structures of strange attractors that may indicate that changes triggered by a given value of a parameter might contribute to an undesired phase shift (transition) and thus consequently impact the system's stability (the system can become chaotic). There are several techniques discussed by Frazier and Kockelman (2004) as well as Ramasubramanian and Sriramm (2002) that can be used to predict which self-shaping transitions may result in the system to become chaotic. In our simulation environment we aim to detect such undesired self-shaping scenarios. The most commonly known is *the largest Lyapunov exponent* technique that can help to predict a chaotic behaviour in the system. The value of the largest Lyapunov exponent is a measure of the rate of nearby trajectories convergence/divergence in each dimension. As the system reshapes (evolves), the sum of a series of attractor values will diverge or converge. If the largest Lyapunov exponent exceeds zero then the system is to become chaotic. To find the largest Lyapunov exponent (Frazier and Kockelman 2004), we can apply the following function:

$$\lambda_{\max} = \frac{1}{N\Delta t}\sum_{t=0}^{N-1}\ln\left(\frac{|s(t+\Delta t)-s'(t+\Delta t)|}{|s(t)-s'(t)|}\right) \qquad (8)$$

Theoretically, with Δt increases, the exponent value converges to its true value. In reality, this may not be always the case as the noise and finite number of data sets determine the range of exponent values. The second technique that may be used to test for chaotic tendencies in the system is the Fractal Dimension Indicator method. For chaotic systems the indicator value will be simply a non-integer. In his work Abarbanel (1996) indicated that there are many kinds of fractal dimensions, but the most practical one is the correlation dimension. Another technique (Frazier, Kockelman, 2004) to test for the presence of chaos uses the Fourier power spectrum function. For periodic data, the system characteristic frequencies of the power spectrum will spike while for the rest of frequencies it will be close to zero. For chaotic data, the spectrum will be broadband and have a broad peak.

5. EXPERIMENTATION

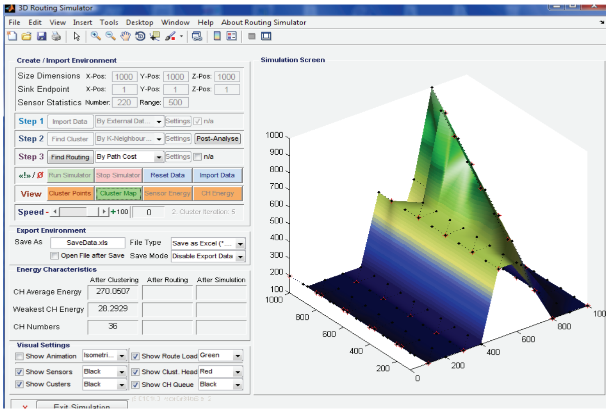
In our simulation experiments we attempt to test the concepts of Bejan constructal theory and verify if the underlying principles can be effectively applied in construction of BIM for a constrained resource system such as WSN. As a guide the Table 1 list analogies between constructal theory and software engineering (architectural design). Following the constructal approach for distributing the "imperfection" in the system, it suggested to seed the more resistive regime (policies) at the smallest scale of the network

At this stage various experimentation has been conducted using the originally developed Matlab based 3D WSN simulator designed for the purpose to test mechanisms of self-organisation and allometric self-shaping. The simulator was designed to handle various stressful conditions in WSN networks. Initial tests of allometric self-organisation in WSNs that follows the Murray's law when forming the 3-dimensional lattice demonstrate favourable patterns of energy usage (see Figure 2, Figure 3, Figure 4 and Figure 5).

Table1: Software Engineering vs Constructal Theory

Software Engineering	**Constructal Concept**
Architectural view, composition	Flow architecture (geometry, structure)
Evolving component, process	Change of structure
System Requirements	Global objective/ constraints
Load balanced architecture	Equilibrium flow architecture
Object or component relation	Fundamental relation
Evolving architectures/components	Non-equilibrium architectures
Removal of design constraints	Increased freedom to morph
Maximisation of data flow in ports	Maximization of flow access

The relations between sensor nodes cross-sectional areas of adjoining links emulate minimal resistance networks according to the generalised Murray's law. On other hand, simulation results of k-Means clustering of WSNs with randomly seeded nodes indicate excessive loss of field energy (Figure 4) and after routing a higher drop-out rate of cluster heads (CH);

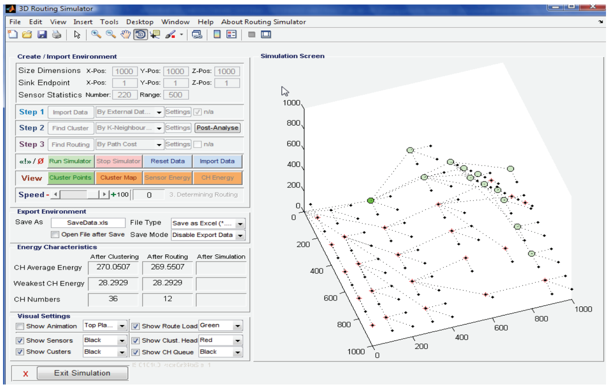


Size Dimensions X-Pos: 1000 Y-Pos: 1000 Z-Pos: 1000
Sink Endpoint X-Pos: 1 Y-Pos: 1 Z-Pos: 1
Sensor Statistics Number: 220 Range: 500
Step 1: By External Dat... ☑ n/a
Step 2: By K-Neighbour...
Step 3: By Path Cost ☐ n/a
Speed: 0 — 2. Cluster Iteration: 5
Save As: SaveData.xls File Type: Save as Excel (*....
☐ Open File after Save Save Mode: Disable Export Data

	After Clustering	After Routing	After Simulation
CH Average Energy	270.0507		
Weakest CH Energy	28.2929		
CH Numbers	36		

☐ Show Animation: Isometri... ☑ Show Route Load: Green
☑ Show Sensors: Black ☑ Show Clust. Head: Red
☑ Show Custers: Black ☑ Show CH Queue: Black

Figure 2: 3D WSN Simulator Testing the Energy Left in The Field After the Process of Clustering (by k-Neighbourhood) in the WSN Organised as a Lattice Structure of 220 Nodes



Size Dimensions X-Pos: 1000 Y-Pos: 1000 Z-Pos: 1000
Sink Endpoint X-Pos: 1 Y-Pos: 1 Z-Pos: 1
Sensor Statistics Number: 220 Range: 500
Step 1: By External Dat... ☑ n/a
Step 2: By K-Neighbour...
Step 3: By Path Cost ☐ n/a
Speed: 0 — 3. Determining Routing
Save As: SaveData.xls File Type: Save as Excel (*....
☐ Open File after Save Save Mode: Disable Export Data

	After Clustering	After Routing	After Simulation
CH Average Energy	270.0507	269.5507	
Weakest CH Energy	28.2929	28.2929	
CH Numbers	36	12	

☐ Show Animation: Top Pla... ☑ Show Route Load: Green
☑ Show Sensors: Black ☑ Show Clust. Head: Red
☑ Show Custers: Black ☑ Show CH Queue: Black

Figure 3: 3D WSN Simulator: Top View of the Lattice Topology After Routing

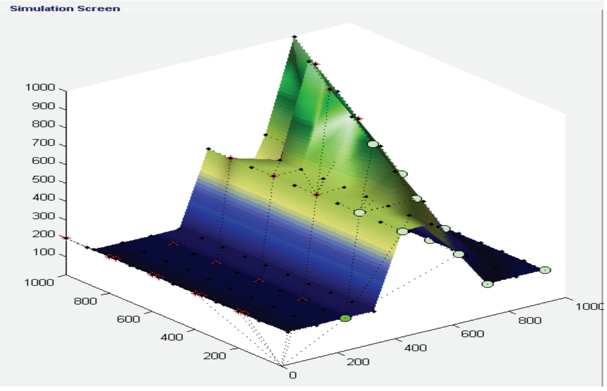
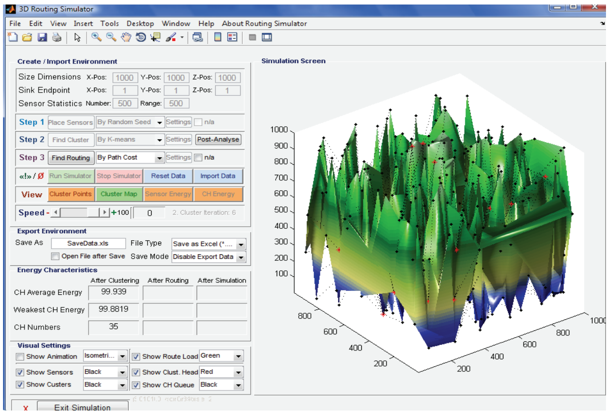


Figure 3: Top View of the Lattice Topology. The Result of Simulation that Tested the Usage of Energy in the WSN when Using the Path-cost Routing Algorithm. White Circles Depict (Energy) Healthy Cluster Heads.

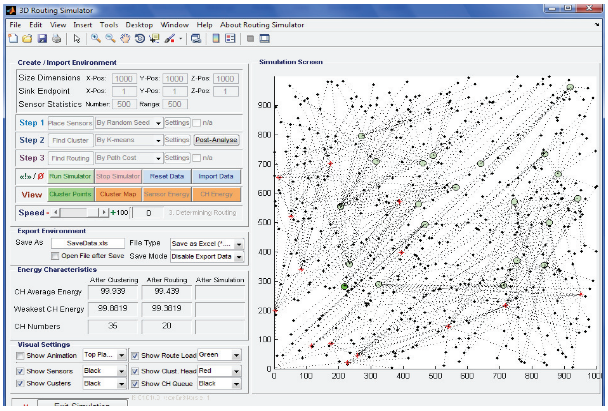

3D Routing Simulator
File Edit View Insert Tools Desktop Window Help About Routing Simulator
Create / Import Environment
Step 1 Place Sensors Settings
Step 2 Find Cluster Settings Post-Analyse
Step 3 Find Routing Settings
«!» / Ø Run Simulator Stop Simulator Reset Data Import Data
View Cluster Points Cluster Map Sensor Energy CH Energy
Speed - +100
Export Environment
Energy Characteristics
Visual Settings
Exit Simulation
Simulation Screen
1000
900
800
700
600
500
400
300
200
100
800
600
400
200
200
400
600
800
1000

Size Dimensions X-Pos: 1000 Y-Pos: 1000 Z-Pos: 1000
Sink Endpoint X-Pos: 1 Y-Pos: 1 Z-Pos: 1
Sensor Statistics Number: 500 Range: 500
Step 1: By Random Seed ☐ n/a
Step 2: By K-means
Step 3: By Path Cost ☐ n/a
Speed: 0 — 2. Cluster Iteration: 6
Save As: SaveData.xls File Type: Save as Excel (*....
☐ Open File after Save Save Mode: Disable Export Data

	After Clustering	After Routing	After Simulation
CH Average Energy	99.939		
Weakest CH Energy	99.8819		
CH Numbers	35		

☐ Show Animation: Isometri...
☑ Show Sensors: Black
☑ Show Custers: Black
☑ Show Route Load: Green
☑ Show Clust. Head: Red
☑ Show CH Queue: Black

Figure 4: Isometric View of the Randomly Seeded WSN.


3D Routing Simulator
File Edit View Insert Tools Desktop Window Help About Routing Simulator
Create / Import Environment
Step 1 Place Sensors Settings
Step 2 Find Cluster Settings Post-Analyse
Step 3 Find Routing Settings
«!» / Ø Run Simulator Stop Simulator Reset Data Import Data
View Cluster Points Cluster Map Sensor Energy CH Energy
Speed - +100
Export Environment
Energy Characteristics
Visual Settings
Exit Simulation
Simulation Screen
900
800
700
600
500
400
300
200
100
0
0
100
200
300
400
500
600
700
800
900
1000

Size Dimensions X-Pos: 1000 Y-Pos: 1000 Z-Pos: 1000
Sink Endpoint X-Pos: 1 Y-Pos: 1 Z-Pos: 1
Sensor Statistics Number: 500 Range: 500
Step 1: By Random Seed ☐ n/a
Step 2: By K-means
Step 3: By Path Cost ☐ n/a
Speed: 0 — 3. Determining Routing
Save As: SaveData.xls File Type: Save as Excel (*....
☐ Open File after Save Save Mode: Disable Export Data

	After Clustering	After Routing	After Simulation
CH Average Energy	99.939	99.439	
Weakest CH Energy	99.8819	99.3819	
CH Numbers	35	20	

☐ Show Animation: Top Pla...
☑ Show Sensors: Black
☑ Show Custers: Black
☑ Show Route Load: Green
☑ Show Clust. Head: Red
☑ Show CH Queue: Black

Figure 5: Top View of the Randomly Seeded WSN After Routing. Only 20 CH Left from the Original 35 Clusters.

6. CONCLUSION

In this paper we have presented the important properties of a Bio-Inspired Middleware model for a resource constrained (WSN) system. Thorough analysis and simulation of such models is required to assess suitability of the approach and a better understanding how the suggested autopoietic properties, principles and rules could help in design and construction of new generation of robust and collaborative Multi-Agent Systems (MAS) (Gorton 2007). The main point is to seed and promote the autonomic and evolving computational properties (abilities) of a WSN system at the local-level and preserve emergence of robustness at the global-level when the system undergoes a major transitional phase(s) change. In order to quantitatively improve BIM model and understand the implications, we are required to study the spatial structure of attractors and the spatio-temporal mappings, cooperative relations, binding energies, agent trajectories and structures of attractors in context of applied biomimetic rules and principles. The spatio-temporal mappings can provide a useful tool to monitor and possibly adjust stochastic properties, relations, connectivity and cooperative relations as well as the interacting agents behaviour in WSN. Simulation of binding energies, agent/data traffic trajectories and structures of attractors can provide information about required parameter values that trigger a change and final impact of phase shifts (transitions) caused by these parameters. Presented simulations can help in establishing whether agent/node movements or traffic trajectory is directed by a positive value of Lyapunov exponent that befalls onto a selected attractor. The technique discussed by Frazier and Kockelman (2004) as well as Ramasubramanian and Sriramm (2002) can help to predict if the system has tendency to become unstable (chaotic behaviour). Similarly to biological systems - the software intensive, real-time network systems such as resource constrained WSN are dynamic in spatio-temporal terms. Various characteristic parameters in these systems are changing with time, thus full understanding of both qualitative and quantitative properties of a BIM using limited capabilities for varying time and topology parameters in simulations is proven to be difficult and unrealistic. These issues need to be realised before dealing with implementations of real systems.

Our study of complex adaptive systems is in its in intermediate phase, so far we can confirm that biomimetic functions based on genetic, immuno-computing and swarm algorithms incorporated in middleware or network management services have important roles to play in the architectural design and optimisation of software-based, resource constrained systems such as WSN systems (Das, Banerjee, Roy 2004). Furthermore, it is expected that our work on MAS-based, biology-inspired frameworks and application solutions would ultimately lead us to design and development of more reliable and flexible systems.

REFERENCES

Abarbanel, H.D.I. 1996. *Analysis of Observed Chaotic Data*. Spinger-Verlag, New York.

Bak, P., 1996. *How Nature Works: The Science of Self-Organized Criticality*. Springer Verlag, New York.

Bejan A., 2000. Shape and Structure, from Engineering to Nature, Cambridge University Press, Cambridge, UK, p.324.

Bejan A., 2002. Constructal Theory of Organization in Nature: Dendritic Flows, Allometric Laws and

Flight", *Design and Nature*, Edited by: C.A. Brebbia & L.J. Sucharov, Wessex Institute of Technology, UK and P. Pascolo, Universita degli di Udine, Italy, Transactio**n:** *Ecology and the Environment* vol. 57.

Bejan A., 2006. Advanced Engineering Thermodynamics, Wiley-Interscience, 2nd edition, 3rd edition, p. 896.

Bonabeau, E., Dorigo, M. and Theraulaz, G., 1999. Swarm Intelligence: From Natural to Artificial Systems, Oxford University Press,

Brown, J. H., West, G. B., & B. J. Enquist 2005. Yes, West, Brown and Enquist's model of allometric scaling is both mathematically correct and biologically relevant. *Functional Ecology* Vol 19, pp.735-738.

Bruzzone, A.G., Longo, F., 2005. Modeling & Simulation applied to Security Systems. *Proceedings of Summer Computer Simulation Conference*, pp. 183-188. July 24-28, Philadelphia, Pennsylvania, USA.

Bunk, S. 1998. Do Energy Transport Systems Shape organisms?" *The Scientist*, December 7. http://www.the-scientist.com/article/display/18316

Calder, W.A., 1984. Size, function and life history. Harvard University Press, Cambridge, Mass.

Chaui-Berlinck, J. G. 2006. A critical understanding of the fractal model of metabolic scaling. *Journal of Experimental Biology* 209:3045-3054.

Clauset, A., Shalizi, C. R. and Newman, M. E. J. , 2007. Power-law distributions in empirical data". http://arxiv.org/PS_cache/arxiv/pdf/0706/0706.1062v1.pdf

Darveau, C. A., Suarez, R. K., Andrews, R. D., & Hochachka, P. W. 2002. Allometric cascade as a unifying principle of body mass effects on metabolism, *Nature,* 417:166-170.

Das, S. K. , Banerjee, N. and Roy, A., 2004. Solving Optimization Problems in Wireless Networks using Genetic Algorithms, Handbook of Bioinspired Algorithms.

Durant, M., 2006. Architecture of optimal transport networks, Physical Review E **73**, 016116 (2006).

Dorogovtsev, S.N. & Mendes, J.F.F., 2003. Evolution of Networks*,* Oxford University Press, Oxford, 2003.

Dressler, F. and Krüger, B., 2004. Cell biology as a key to computer networking, German Conference on Bioinformatics 2004 (GCB'04), Bielefeld, Germany, Abstract and Poster. Inspected on 3-01-2007,www7.informatik.unierlangen.de/~dressler/lectures/sel**b**storganisation-ss05/

Effroni S., Harel D. and Cohen I., 2005, Reactive Animation: Realistic Modeling of Complex Dynamic Systems, *IEEE Computer*, pp.33-46, January.

Eigen, M., and Schuster, P. 1982. Stages of Emerging Life - Five Principles of Early Organization, *Journal of Molecular Evolution*, **19**, v. 47.

Eigen, M., and Winkler, R., 1993, Laws of the Game: How the Principles of Nature Govern Chance. Princeton, N.J.: Princeton University Press.

Enquist, B.J., Kerkhkoff, A.J., Stark, S.C., Swenson, N.G., McCarthy, M.C. and Price C.A., 2007. A general integrative model for scaling plant growth and functional trait spectra, *Nature,* Sept 13, 449(7159):218-22.

Frazier, C., Kockelman K.M., 2004. Chaos Theory and Transportation Systems: An Instructive Example, 83rd Annual Meeting of the Transportation Research Board, January 2004, Washington D.C. www.ce.utexas.edu/prof/kockelman/public_html/TRB04Chaos.pd

Gillooly, J. F. and Allen. A. P., 2007. Changes in body temperature influence the scaling of VO_2_max and aerobic scope in mammals. "Biology Letters", 3: 99-102.

Grant, B., 2007. "The powers that might be," *The Scientist*, March 1st.

Gorton, I, 2004. Evaluating agent Architectures: Cougaar, Aglets and AAA, *Lecture Notes in Computer Science, Volume.* 2940, Springer New York, 264-274
http://www.the-scientist.com/2007/3/1/42/1/

Gustavsson, R. & Fredriksson M., 2003. Sustainable Information Ecosystems, *Lecture Notes in Computer Science*, Volume 2603/2003, Springer Berlin / Heidelberg, 123-138.

He, Ji-Huan, Chen H., 2003. Effects of Size and pH on Metabolic Rate, *International Journal of Nonlinear Sciences and Numerical Simulation* 4, pp. 429-432.

He, Ji-Huan, Zhang J., 2004. Fifth Dimension of Life and the 4/5 Allometric Scaling law for the Human Brain. *Cell Biology International*. 28(11), pp. 800-15.

Hull, B., Jamieson, K. and Balakrishnan, H., 2004. Mitigating Congestion in Wireless Sensor Networks, *Proceedings of the 2nd International Conference on Embedded Networked Sensor Systems*, ACM SenSys 2004, Nov., pp. 134-147.

Ishida, Y., 1997. The immune system as a prototype of autonomous decentralized systems: an overview," *In proceedings of 3rd International Symposium on Autonomous Decentralized Systems* (ISADS 97).

Kauffman, S.A., 1993. The origins of Order, *Oxford University Press*, Oxford.

Kaufmann, S.A., 1995. At Home in the Universe: The Search for the Laws of Self-Organization and Complexity. Oxford University Press, New York.

Keele, J.W. and Wray J.E. , 2005. Software Agents in molecular computational Biology*, Briefings in Bioinformatics*, Vol.6, No.5, December, 370-379.

Kennedy, J., 2006. Swarm Intelligence, in *Handbook of Nature-Inspired & Innovative Computing,* Editor: A. Zomaya, Springer Verlag, New York, pp.187-221.

Kleiber, M., 1932. *Body size and metabolism*, Hilgardia, 6 pp.315-53.

Kozlowski, J., Konarzewski, M. & Gawelczyk, A. T., 2003. Cell size as a link between non-coding DNA and metabolic rate scaling. *Proc Natl Acad Sci U S A* 100:14080-14085.

Kozlowski, J. & Konarzewski, M., 2004. Is West, Brown and Enquist's model of allometric scaling mathematically correct and biologically relevant? *Functional Ecology* 18:283-289.

Krishnamurthy, E. V. et al, 2006a. Multiset Rule-based Programmming Paradigm for Soft Computing in Complex Systems*,* in *Handbook of Nature-Inspired & Innovative Computing,* Ed: A. Zomaya, Springer, New York, pp. 77-109.

Krishnamurthy, E.V. & Murthy, V. K., 2006b. Distributed Agent Paradigm for soft and hard computation*, Journal of Network and Computer Applications*, Vol. 29, 124-146.

La Barbera, M., 1990, Science **249**, 992 (1990)

Livingston, I., 2006. *Between Science and Literature: An Introduction to Autopoietics*. University of Illinois Press. —an adaptation of autopoiesis to language.

Loureiro, A.A.F. and Ruiz, L.B., 2007. Autonomic Wireless Networks in Smart Environments, In Proceedings of the 5th Annual Conference on Communication Networks and Services Research, CNSR '07. Fredericton, New Brunswick, Canada.

Makarieva, A. M., V. G. Gorshkov, & B. L. Li. 2005. Revising the distributive networks models of West, Brown and Enquist (1997) and Banavar, Maritan and Rinaldo (1999): Metabolic inequity of living tissues provides clues for the observed allometric scaling rules. *Journal of Theoretical Biology* 237:291-301.

Maturana, H. & Varela, F., ([1st edition 1973] 1980). *Autopoiesis and Cognition: the Realization of the Living*. Robert S. Cohen and Marx W. Wartofsky (Eds.), Boston Studies in the Philosophy of Science **42**. Dordecht: D. Reidel Publishing Co. ISBN 90-277-1016-3 (paper) — the main published reference on autopoiesis

Maturana, H. R. & Varela, F. J. (1987). *The tree of knowledge: The biological roots of human understanding*. Boston: Shambhala Publications.

McMahon, T. A. and Bonner, J. T., 1983. On Size and Life. Scientific American.

Mitzenmacher, M., 2003. A brief history of generative models for power law and lognormal distributions". *Internet Mathematics* **1**: 226–251. http://www.internetmathematics.org/volumes/1/2/pp226_251.pdf

Niklas, K. J. 1994. Plant allometry: The scaling of form and process. University of Chicago Press, Chicago.

Newman, M.E.J., 2005. Power laws, Pareto distributions and Zipf's law. *Contemporary Physics* 46: pp. 323–351. Inspected on 12-03-2007,http://www.journalsonline.tandf.co.uk/openurl.asp?genre=article&id=doi:10.1080/00107510500052444

Phillips M.L., 2006. "Study challenges metabolic scaling law," *The Scientist*, January 26. http://www.the-scientist.com/news/display/23012/

Prigogine, I., (ed.) et al., 1993. Chaotic Dynamics and Transport in Fluids and Plasmas, Springer Verlag.

Ramasubramanian, K. & Sriramm, M.S., 2002. A Comparative Study of Computation of Lyapunov Spectra with Different Algorithms, Accessed March 27, 2008.
http://xxx.lanl.gov/PS_cache/chaodyn/pdf/9909/99090pdf

Reis, A.H., Miguel A.F., Aydin M., 2004. "Constructal theory of flow architecture of the lungs", *Journal of Medical Physics*, May 2004, Volume 31, Issue 5, 1135-1140.

Resconi G. and Chaczko Z., 2007. Ego-Morphic Agent Theory, 8th International Conference CASYS'07 on Computing Anticipatory Systems, August 6-11, 2007, Liège, Belgium.

Wan, C.Y. , Eisenman, S. B. and Campbell, A. T. 2003. CODA: Congestion Detection and Avoidance in Sensor Networks, Proceedings of the 1st International Conference on Embedded Networked Sensor Systems, ACM, SenSys 2003, pp. 266-279.

Wang, C., Sohraby, K. and Li., B, 2005. "SenTCP: A hop by hop congestion control protocol for wireless sensor networks," IEEE INFOCOM'05.

West, G.B., Savage, V.M., Gillooly, J., Enquist, B.J., Woodruff, W.H. & J.H. Brown, 2007. Response to Darveau et al.: Why does metabolic rate scale with body size?, *Nature,* 421:713.

West, G.B & Brown J.H. 2005. "The origin of allometric scaling laws in biology from genomes to ecosystems: towards a quantitative unifying theory of biological structure and organization," *J. Exp. Biol.* 208: pp. 1575-1592, 2005. http://www.the-scientist.com/pubmed/15855389

Krause A., Guestrin, C., A. Gupta A., Kleinberg, J., 2006. Near-optimal sensor placements: Maximizing information while minimizing communication cost. In Proceedings of Information Processing in Sensor Networks.

AUTHOR BIOGRAPHY

Zenon Chaczko completed a B.Sc. in Cybernetics and Informatics in 1980 and a M.Sc. in Economics in 1981 at the University of Economics, Wroclaw in Poland., as well as completed MEng in Control Engineering at the NSWIT 1986, Australia. For over 20 years Mr Chaczko has worked on Sonar and Radar Systems, Simulators, Systems Architecture, Telecommunication network management systems, large distributed Real-Time system architectures, network protocols and system software middleware. Mr Chaczko is a Senior Lecturer in the Information and Communication Group within the Faculty of Engineering at UTS.

CARDIOSCOPE SIMULATOR SYSTEM

Zenon Chaczko and Hany Shehata

ICT Group,
Faculty of Engineering, University of Technology, Sydney, Australia

zenon.chaczko@uts.edu.au, Hany.E.Shehata@eng.uts.edu.au

ABSTRACT

There is a need for effective, reliable and economic diagnostic technique to detect heart problems at an early stage. This paper presents an innovative approach to the ECG diagnostic modeling that considers a representation of the Inverse problem in multi-dimensional space. Assumptions are made regarding the shape and values for the electrical sources inside the human heart in order to estimate the body surface electrical potentials. An analytical set of expressions used in the proposed diagnostic model combined with the effect of associated source parameters is discussed, analysed and verified. The paper introduces an innovative Inverse problem method for determining the inner heart electrical activity parameters. Results can be then visualized given availability of a stream of body surface potential data. WSN technology is to be applied for collecting and processing diagnostic data.

Keywords: ECG diagnostic model, cardioscope, simulation model

1. INTRODUCTION

The ECG is considered to be one of the oldest (Einthoven 1908), and most reliable techniques for detection of heart abnormalities. Inevitably, it is one of very few techniques that can also be used for construction of predictive diagnostic models of the electrical activity of the heart. The reliability of ECG method depends on two important factors:

1. Quality and accuracy of statistical data that allows accumulation and correction of ECG shapes with types of diseases causing it. The ECG involves recording electrical signals measured from human body and reflecting electrical activities of the heart. Also, the method can be used to calculate characteristic parameters of the heart from captured electrical signals at the human body.
2. Cardiologists' experience and diagnostic skills in analysing heart problems using ECG technique. The training of cardiologist takes time. There are recorded cases (and undoubtedly, there will be such cases in the future) which impose critical situations, where a doctor's indecision is intolerable as only a decision taken fast can save patient's life.

For the reasons stated above, the research studies of the heart functions should extensively stress the need for modeling, analysis and diagnosis of the heart problems taking not only predominantly medical but also an engineering point of view. Two complementary techniques that use mathematical apparatus in construction of electric models contribute to formulation of ECG solution. The first technique allows building an abstract model representing the system and then examines the effect of the assumed parameters. This technique of computing ECG is also known as the forward problem. The second technique, which is based on the forward problem, is termed the inverse problem. It stars from available body surface potential data and ends up with recognizable electrical activities that can help in diagnosing heart problems. In the literature, both techniques are equally well covered. In our study, the emphasis is on the inverse problem. Since both problems are complementary, it is necessary to include some analysis of the forward problem. The introductory part will provide a brief description of the electric activity of the heart including the internal sequences of excitation and an overview of the system used to measure the ECG. The analytical stage is divided into three sections: (1) a calculation model for measuring body surface potentials, (2) collection of readings from the chest leads connectors (as originally postulated by Einthoven) and (3) the image reconstruction of the heart dipole vector that can be used to study the nature of ECG waves for diagnostic purposes.

This paper presents an innovative and simplified approach to the ECG diagnostic modeling that considers the representation of the Inverse problem in multi-dimensional space (in this study 2D only). Firstly, an analytical set of expressions for the proposed simplified model is elaborated on, and then the effect of source parameters is verified and analysed. Finally, a new approach to solve the inverse problem itself is presented. The process of Fourier series analysis helps to extract information about the source parameters embedded in the surface potential data. These parameters help to determine the electrical source width angle in a real time.

The main aim is to provide an improved model for the human body for ECG purpose. The model is used to estimate the body surface electrical potential. Assuming a certain shape and value for the electrical source inside the human heart we calculate the body surface potential. The research also aims to introduce a method for determining the inner heart electrical activity parameters and displaying them given a stream of body surface potential data. Current development involves design and deployment of WSN technology for collecting and processing potential data.

2. GENERAL ASPECT OF ECG (ELECTROCARDIOGRAPHY)

Active tissue during activity produces electric currents in the body. So, the heart acts as an electric generator when it beats. The electrical state of the heart varies during the cardiac cycle. This variation takes the form of a wave of electronegative on the outer surface of myocardial member (i.e. depolarisation). These electrical changes are similar to those in skeletal muscles and can be recorded by a sensitive galvanometer. ECG is a record of the electrical changes in the heart during any cardiac cycle. In animals recording is done by putting electrodes directly on the heart surface. In human, blood and tissue fluids are relatively good conductor of electric current. So, human ECG is recorded by putting two electrodes on the skin. The particular arrangement of the two electrodes is called a lead.

2.1. Body Surface Potential And The Lead Theory

The conventional electrocardiogram is a time base record of the potential differences developed in one or more leads (a lead is a combination of at least two electrodes). It started by recording the potential differences between the left arm, right and the left leg which was described by Einthoven early in the nineteenth century. These were originally called the limb leads I, II and III. Each of them refers to the potential difference between one of the three points and the electric mid-point of the other two points. Six additional pericardial chest leads were introduced by (Wolf Worth and Wood 1932), to provide more information about the electric activity of the heart. These are the potential difference between central terminals of Wilson (the average of the three points on the chest). These set of leads records the myocardial activity in the horizontal plane. It is important to note that an individual chest lead does not represent the electrical potential of a localised area of the underlying myocardium. But it represents all the electric events all over the heart as viewed from that particular lead site. However, owing to the proximity of the pericardial lead to the surface of the heart, potentials generated in the underlying portion of the heart muscle will dominate other relatively more remote portions.

2.2. Chest Leads

Six electrodes are placed at six different positions around the chest. These six leads are monitored from six progressively different positions and they are numbered V1 to V6 in successive steps. If leads V1 to V6 are assumed to be the spokes of wheels. The centre of the wheel is the (A-V) node. These six leads are cutting the plane of the body into top and bottom halves in horizontal plane. In these leads we show the complex QRS is mainly positive or negative.

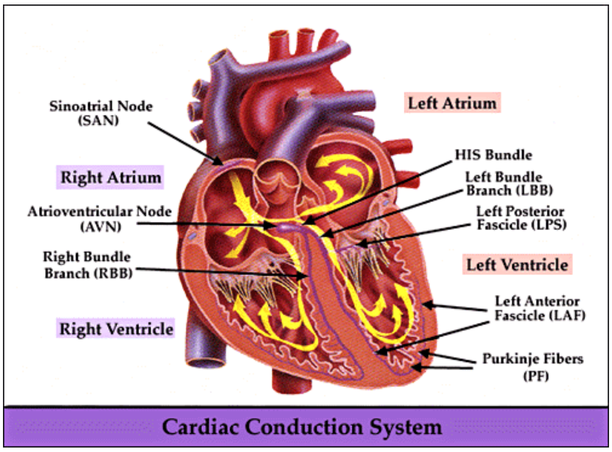

Sinoatrial Node (SAN)
Left Atrium
HIS Bundle
Left Bundle Branch (LBB)
Right Atrium
Atrioventricular Node (AVN)
Left Posterior Fascicle (LPS)
Right Bundle Branch (RBB)
Left Ventricle
Left Anterior Fascicle (LAF)
Right Ventricle
Purkinje Fibers (PF)
Cardiac Conduction System

Figure 1: Cardiac conduction system.

3. RECONSTRUCTION OF THE HEART DIPOLE VECTOR

Until now, Einthoven's triangle has been widely accepted for clinical applications as a basic relationship between the bipolar electrocardiogram leads and the heart vector. It means that bipolar leads are simple projections of the cardiac vector. Although this concept is practically widely accepted, it includes the following assumptions:

1) The electric activity of the heart is represented at any instant by a single dipole.

2) The human body is regarded as an imaginary sphere within infinite conductor, with the cardiac dipole located at its centre and the electrodes located at equilateral triangle on the imaginary sphere.

3) The electric conductivity of the body is uniform (homogeneous) and independent of the direction of the cardiac vector (isotropic)

In 1964 Burger and Van Milaan, started with general of the lead potential V as follows:

$$V = \vec{C} \cdot \vec{P} = C_x P_x + C_y P_y + C_z P_z \qquad (1)$$

Where: $\vec{P}$ is the cardiac dipole vector and $C_x, C_y \,\&\, C_z$ are coefficients correspond to the site of measuring electrode on the body surface. The coefficients of the position vector were calculated for each point of interest on the body surface. For the designated medium and dipole position, each surface point has its space vector C which is constant. Therefore, according to the equation the potential V of that point can be obtained by projecting the time-varying cardiac vector P on the fixed vector C and then multiplying it by the magnitude of C. In this sense each point on the body surface corresponds to a vector C and the end of it (tip point) represents an imaginary surface known as the image surface. The connections between the point corresponding to the right arm (RA), the left arm (LA) and the left foot (LF) on the image surface from a non equilateral triangle in a plane somewhat oblique to the frontal plane. This triangle is known as the Burger triangle.

3.1. The Normal ECG

This consists of the three positive waves above the isoeletric line P-R and two negative waves below isoeletric line Q and S, where P is atrial in origin and Q R S T are ventricular. These waves were discovered by Einthoven in 1895.

1) P-wave represents spread of excitation wave in both atria i.e. depolarization of the atrial muscle. It is a small positive wave starts 0.02 sec before the mechanical response of atrium.

2) Q.R.S. represents spread of excitation wave in ventricles i.e. depolarization of ventricular muscle. It begins 0.02 sec before mechanical response of ventricular.

3) Q-wave (0.02s) is small negative wave representing spread of excitation wave in interventricular septum.

4) R-wave (0.04s) is the largest wave. It represents the excitation of the apex of ventricular walls and the base of ventricles.

5) S-wave (0.02s) is a small negative wave representing retreat of excitation of the remaining part of the base ventricles.

6) T-wave (0.25 s) is a positive wave representing retreat of excitation wave from ventricular muscle i.e. repolarisation of ventricles.

3.2. Problem Formulation

The body surface potentials are directly related to the electric source within the heart. Therefore, any abnormality in the cardiac source and/or in the electrical properties of the body torso is totally reflected on the ECG pattern. Various studies focused on the relationship between cardiac sources and the body surface potential. With the use of mathematical and electrical models we have two complementary techniques. The first one is to assume a certain model representing the source, to try to calculate the body surface potential from this assumed source and to examine the effect of the assumed parameters on the computed ECG. This is called the forward problem. The second one is based on the forward problem. It starts from available body surface potential data and trying to estimate electrical sources in the heart and causes for having data in the forward problem. The forward solution is affected by several parameters such as body shape, heart shape, body size, heart size, blood conductivity, lunges size, shape and conductivity. So, because of the complexity of the problem, the modeling (simulation) should include some simplifications. This is done by eliminating the effects of some parameters, or giving less weight to others. While stressing on another parameters assuming that their effect is dominant. According to the type of simplification introduced, two main types of modeling were considered.

1. A homogeneous with realistic body and heart shapes which is a quantized form model.
2. Analytical inhomogeneous model using special geometrical shapes to approximate the human body and the heart shapes.

The simplification of the first type neglects the effect of inhomogeneity while the second type approximates the body and heart shapes. In this paper we are first considering a very simplified model represented in two dimensions. That offers an analytical expression for the proposed simplified model. It describes the surface potentials according to the new model. While in the second part we propose a different approach to solve the inverse problem. Solution for the inverse problem is not as a simple task for reasons such as:

1. Complex mathematical models need to be considered.
2. Simulation studies have not yet established acceptable parameters (numbers) for the model.
3. The solution of the problem does not offer unique results. The formulation of the problem itself has not yet reached a point where it could be solved on one to one basis.

For all these reasons alone we could confidently say that a break-through to the inverse problem does not exist until now. In this paper we try to propose an approach for a solution to the inverse problem. First, the model considered is a simple two dimensional concentric circular model with a homogeneous body. We used a trial and error technique to solve the inverse problem. That technique uses the body surface potential data to extract the required information about the source.

It is important to note that the difficulty of solving such a problem is obviously figured out if we knew that it is not a one to one problem. The proposed solution thus far has not been proven to be unique.

4. ANALYTICAL SOLUTION OF THE 2-D CONCENTRIC CYLINDERS MODEL

4.1. The Single Layer:

The source of all the electrical activities of the heart was first considered as a single dipole located at the heart centre as stated by Einthoven. The complicated surface distribution of the activation wave of the electric cardiac source could be simplified to have the following characteristics:

1. Discontinuity of the voltage with the value of its strength.
2. Continuity of normal current. The voltage at any point inside the heart was found to be given by V1 and outside the heart surface is V2

$$V_1(r,\theta)=\sum_{n=1}^{\infty}-\frac{M\theta_\circ \sin(\frac{n\theta_\circ}{2})}{4\pi(\frac{n\theta_\circ}{2})}(\frac{r}{r_\circ})^n \cos(n\theta)-\frac{M\theta_\circ}{2\pi} \qquad (2)$$

$$V_2(r,\theta) = \sum_{n=1}^{\infty} - \frac{M\theta_\circ \sin(\frac{n\theta_\circ}{2})}{4\pi(\frac{n\theta_\circ}{2})} (\frac{r_\circ}{r})^n \cos(n\theta) \qquad (3)$$

4.2. Potential Distribution In The Radial Direction

The potential distribution suffers from attenuation with the increase of r. This attenuation increases gradually from heart surface .reaching its maximum value at the body surface. Figure2 shows the potential distribution across the human body from the heart surface to the body surface. It is plotted as calculated from the above formulas. We assumed b =0.4 and $\theta_\circ = 90^\circ$ (see 4.3)

4.3. Parameter Estimation By Using Fourier Series:

The main reason for choice of a two dimensional model is the form of the solution which could be carried out in a sinusoidal harmonics. The sinusoidal harmonics are very easy to handle.

Assuming that b (heart to body radius ratio) and $\theta_\circ$ (heart electrical wave width angle) are known, we solve the forward problem using the double layer model (similar to the single layer with assuming finite body radius r1)

.

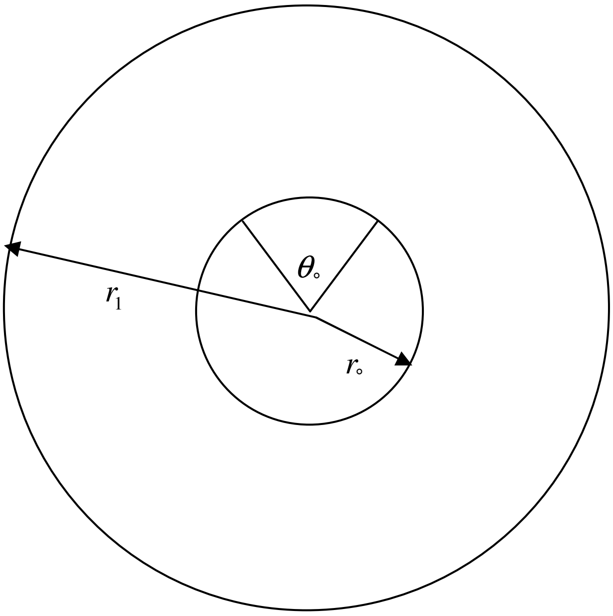


Figure 2: 2-D Model of The human body and heart

However, b is not known and hence solving the inverse problem reduces to finding a method to estimate b and $\theta_\circ$. Using two terms of Fourier series expansion of $V_2(r_1,\theta)$ and an initial estimation value for b, we used a trial and error method to calculate b and $\theta_\circ$ (to be discussed next section). We have plotted a number of curves to clarify the range that should be used for estimating the value for b. For b=0.7 the trial and error method is to be repeated for different values of harmonic numbers (n). The error in estimation of $\theta_\circ$ is calculated. The presented table summarizes the final results. From these results it can be concluded that the actual values of b and $\theta_\circ$ are nearly reached after five iterations of the loop.

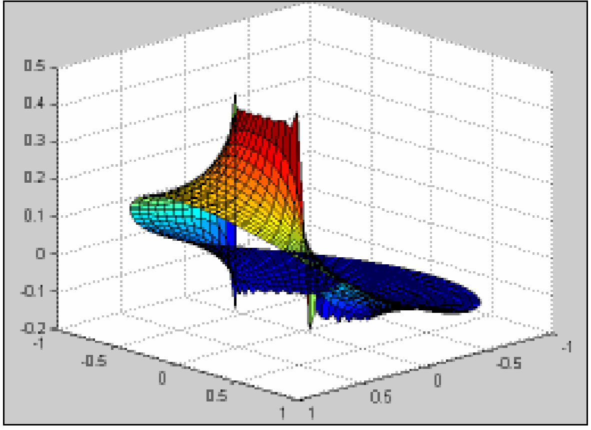


Figure 3: Potential distribution from the heart surface to the body surface.

5. EXPERIMENT RESULTS

5.1. Angular Variation Of Radius Ratio b

We have examined the effect of giving ***b*** a value other than its actual value in the solution of the inverse problem. We have also compared the actual figures at $b = b_{actual}$ and the estimated figures at $b \neq b_{actual}$ The comparison gives interesting results which can help to clarify that choosing a guess value for ***b*** that is larger than b_{actual} is closer to the correct solution than choosing a value of ***b*** smaller than b_{actual} (e.g. we can always start by assuming a value for ***b***=0.7) The trial and error method is repeated for different values of harmonic numbers *(n)*. Each time using the same value for ***b*** we calculated the error in estimation of $\theta_\circ$ for each value of n. We also repeated the above for different values of $(b,\theta_\circ)_{actual}$. The table shown summarizes the results.

Table1: Computations the radius ratio b.

Actual		No. of loops					Error in result				
b	$\theta_\circ$	1	2	3	4	5	1	2	3	4	5
0.3	60	2	18	3	3	1	X	0	1	1	1
	90	18	3	1	1	18	X	1	0	0	X
	120	18	1	1	3	1	0	0	0	0	X
	150	2	3	18	3	18	0	0	X	X	X
0.4	60	1	18	3	2	1	X	0	1	1	1
	90	18	2	1	1	18	X	1	0	0	X
	120	18	1	1	3	1	0	1	0	X	X
	150	18	3	18	1	18	0	0	X	X	X
0.5	60	2	6	2	2	1	X	0	1	1	1
	90	18	2	1	1	18	X	1	0	0	X
	120	4	1	1	3	1	X	1	0	X	X
	150	18	3	18	1	12	0	0	X	X	X
0.6	60	2	18	2	2	1	X	0	1	1	1
	90	18	2	1	1	18	X	1	0	0	X
	120	18	1	1	2	1	X	1	0	X	X
	150	2	3	18	1	18	0	0	X	X	X

5.2. Real readings of body potential versus time

Figure 4 depicts the real body surface data versus time. The normal ECG wave can be recognized along one horizontal axis while the body potential distribution is shown along the other horizontal axis as measured from the six chest leads.

From the obtained results we can conclude that the actual values of $(b,\theta_\circ)$ are nearly reached after five times of the loop when we take n = 2 or n=3.

6. CONCLUSION

In this paper we have presented a solution of the inverse problem applicable for ECG diagnostics. It is possible fuse the electrical wave data with real time audio and/or video stream that is to be fed to a monitoring and/or recording device. Further development on the basis of the above study can be progressed by be the deployment of WSN (Wireless Sensor Network) technology that can collect the body surface potential from random and continuous readings. This technology can provide a rich source of analytical information.

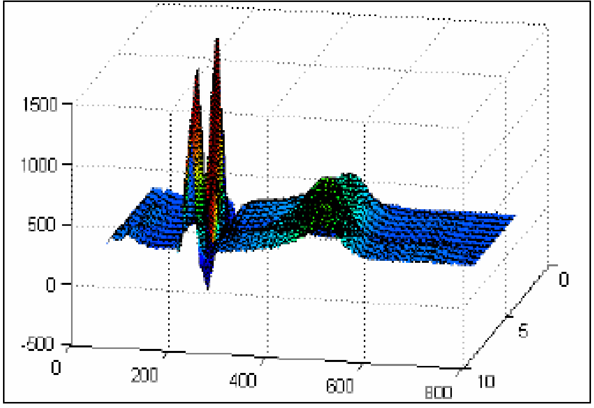


Figure 4: The wave of potentials across a body in time.

REFERENCES

Einthoven, W., Fahr, G., Waart, A., 1950. *On the direction and manifest size of the variations of potential in the human heart and on the influence of the position of the heart on the form of the electrocardiogram Arch. Physiol*150: 275.

Gulrajani, R. M. , Savard, P. , Roberge, F. A., 1988. *The inverse problem in electrocardiography: Solution in terms of equivalent sources.* CRC *Crit. Rev. Biomed. Eng*16: 171-214

Gulrajani, R. M. , 1989. *The inverse problem of electrocardiography. In Comprehensive Electrocardiology. Theory and Practice in Health and Disease*1:237-88.

Burger, H. E., Milaan, J. B., 1948. *Heart Vector and Leads*, *Brit. Heart* 10:229.

Jenkins, D, 1996. *ECG time line - History of the electrocardiogram*, Available from: http://www.ecglibrary.com.ecghist.html [May 2008].

McSharry, P. E., Clifford, G.D., Tarassenko, L., Smith, L., 2003. *A dynamical model for generating synthetic electrocardiogram signals. IEEE Transactions on Biomedical Engineering.* 50:289-294.

Medvegy M., Duray, G., Pintér, A., Préda, I. 2002. *Body Surface Potential Mapping: Historical Background, Present Possibilities, Diagnostic Challenges, In Annals of Noninvasive Electrocardiology*, 7:139.

Throne, R. D., Olson, L.G., 2000. *Fusion of body surface potential and body surface Laplacian signals for electrocardiographic imaging Biomedical Engineering. IEEE Transactions* 47:252-462.

Wolferth, C. C, , Wood, F. C., 1932. *The electrocardiographic diagnosis of coronary occlusion by the use of chest leads. Am J Med Sci* 183:30-35

AUTHORS BIOGRAPHY

Zenon Chaczko

Senior lecturer in University of Technology Sydney. He has many publications and lectures in ICT and software Engineering. He is the research supervisor of many Master of Engineering students. He is teaching ICT in post graduate level. **Zenon** also is the supervisor of many under graduate capstone project.

Hany Shehata

BSc in Telecommunication Engineering from Alexandria University Egypt. Master of Engineering Studies in Software and Information Systems Engineering from University of Technology Sydney UTS. **Hany** is a research student in UTS and he has several years of experience in the ICT industry. He has some efforts in the bio engineering research with strengths in mathematics and modeling.

A SMART-SHOP SYSTEM - MULTI-AGENT SIMULATION SYSTEM FOR MONITORING RETAIL ACTIVITIES

Zenon Chaczko[a], Christopher C. Chiu[b]

[a]Zenon Chaczko, ICT Group, Faculty of Engineering, University of Technology, Sydney, Australia,
[b] Christopher Chiu, Faculty of Engineering, University of Technology, Sydney, Australia

[a]zenon.chaczko@uts.edu.au, [b]christopher.c.chiu@uts.edu.au

ABSTRACT

This paper outlines a concept of employing intelligent agents to simulate and deploy a Smart-Shop System for retail environments. The discussion will examine the technical and conceptual challenges, main ideas and the final design rationale. The design model of the Smart-Shop System uses Multi-Agent Simulation Systems (MASS) and distributed middleware frameworks to automate a wide variety of asset monitoring and control tasks in traditional and distributed system concerns, which can be applied to similar application areas. Retail organisations containing any form of physical inventory can benefit from the proposed software solution by allowing assets to be efficiently managed and monitored. This would improve the quality of business trade and strategic marketing to sustain sales growth.

Keywords: Information Technology, Intelligent Agents, Inventory Management, Retail Commerce, Software Engineering

1. INTRODUCTION

With the advances in ICT technology, modern inventory control methodologies employed in retail environments require manual verification by sales staff when performing periodic stock-takes. The hidden labour costs of inventory control, such as locating inaccessible items, returning misplaced items to their original locations and basic theft prevention tasks such as bag inspections can result in decreased human resource efficiency to assist the individual needs of customers.

The development of autonomous wireless sensors in the past decade, in terms of cost, performance and component size, has reached a stage where practical technological applications in retail facilities has become a feasible option in monitoring inventory. Meanwhile, the cost of labour continues to increase due to economic conditions, coupled with the training costs of hiring new sales staff, such as natural attrition, all add to the overall cost of providing customer rapport in retail environments (Buckland 2006). Therefore, inventory control tasks, such as stocktaking, resorting and theft prevention; increase proportionately with labour costs as the size of a retail store's inventory increases.

The implementation of software intelligent agents in simulating support service scenarios is an evolving software engineering concern. This research investigation is designed to model various concerns in retail environments to demonstrate the feasibility of using intelligent agents to assist asset management decisions in commercial organisations. By determining the essential requirements to build a framework which the simulation will operate, including retail space, inventory quantity and the headcount of customers and retail staff; the project aims to simulate the practical application of using wireless sensors in retail environments with the coupling of traditional monitoring systems such as closed-circuit television, entry gates and various check points.

The theoretical potential of wireless sensor networks (WSN) can be viewed beyond the traditional inventory control technology domain, such as bar-coding and RF security sensors (Blackwell 2003; Buckland 2006), as wireless technologies are not bound by line-of-sight constraints. The project aims to develop a practical implementation of a sensor network environment that can be simulated with software intelligent agents in the scope of a retail organisation.

2. EXAMINING INVENTORY SYSTEMS IN RETAIL ORGANISATIONS

The development of novel systems to meet the special needs of retail environments can be built around existing industry knowledge of sensor network technologies, and driven by scientific initiatives for Smart-Systems in retail organisations. The demand for the implementation and development of such systems cannot always be met by commercial software enterprises, so a considerable interest is needed to appoint local staff to manage and direct the effort. However, these staff would have limited experience in designing, coding, configuring, interfacing and installing traditional systems and would not have management experience. Thus an emphasis in training and modelling the simulation should be a priority in the final model, for retail operators to provide feedback as to the authenticity of the simulation model.

The fundamental concepts to be examined in realising the Smart-Shop System are as follows (Blackwell 2003; Buckland 2006):

- The 'Smart-Shop' refers to the store's automation of procedures. The topic scope is the context of a retail store with distinct, physical collections.
- Stock records can be data-intensive as inventory varies, as the store may need to store unique physical or component attributes of the inventory.
- Store records must be concerned with specific quantities of merchandise; a store that has inaccurate stock control will not function efficiently
- Each product record contains unique details of each item, including the quantity and item particulars, including serial numbers, brand, model number and price (both wholesale and mark-up values).

For these main concerns, the automation of record-keeping and asset management in a retail store reaffirm the need to ensure that records remain accurate and consistent. Furthermore, the automation of record-keeping with a perpetual inventory system allows the decentralised access to records. Retail staff and customers can verify the stock quantity and its location without being physically present through visual verification. The cost trends of technology and labour consistently show how computerised automation is most effective when tasks are tedious and monotonous in nature, as shown in Figure 1. Tasks such as physically locating merchandise on the shelf, and ensuring the correct sorting order, are typical examples of manual procedures accomplished in retail environments that are prone to error.

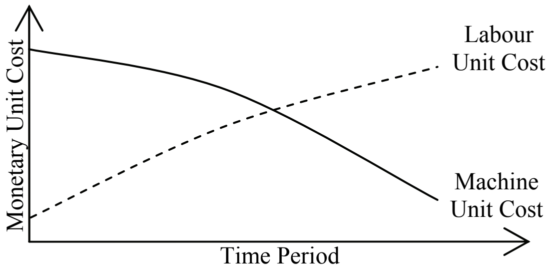


Figure 1: Labour and Technology Trends (Buckland 2006)

The common assumptions made in the context of modern technology to plan ahead for future concerns are listed below (IBM Corp. 2006; Blackwell 2003):

1. Computational Efficiency: Improved computing performance leads to lower cost of ownership. This increases the accessibility and maintainability of computer technology by the end-user.
2. Automated Inventory Monitoring: Wireless sensor network technologies will become more ubiquitous and deployable in confined or restricted spaces, leading to improved optimisation and deployment. Existing bar-coding technologies requires line of sight and require manual operation.
3. Data Storage Capacity: Storage costs expressed as per unit cost decrease as data capacities of storage mediums increase. Data access rates to storage mediums continue to improve over time.
4. Labour Optimisation: Labour costs continue to increase as a result of economic factors, principally with inflation. Manual and repetitive tasks are most prone to inaccuracy, which can lead to inefficiency.

Providing customer services in a typical retail store continues to be labour intensive, with up to two-thirds of a retail budget expenditure accounting for human resources. The purpose of simulating a Smart-Shop System is to demonstrate the feasibility of modelling a retail environment to examine real-world concerns through multi-agent systems. The project aims to gather and structure the simulated data through rudimentary data mining techniques to determine the information that is most suitable for aiding the business decisions of the retail organisation.

The practical implications for implementing the system will be based on the final project system implementation. The main focus is the feasibility of using the simulated data to convey the information to the business operators in such a fashion that would influence future planning decisions, as such optimising operational efficiencies that reducing capital tied up in inventory.

3. SMART-SHOP SYSTEM – THE SOLUTION CASE STUDY

The main concerns of the project are structured so that their content is not interdependent on one another, but are complimentary in nature, delivering a cohesive set of functionalities once completed. The component concerns are examined in detail (Jacobson 1998):

3.1. Review of Underlying Technologies

To examine hardware technologies used in the Smart-Shop System through wireless sensor networks. The nature of sensor networks lends itself to employ a distributed software middleware infrastructure that would interconnect the simulation logic and the data processing modules, thus forming the basis of the technical implementation of the system architecture.

3.2. The Development Methodology Employed

The Software Analysis and Design methodology examined in the scope of the project is based on the iterative development model. The basis for implementing the iterative model allows for the active participation of the business operators to review the modelling scenario, such as to fit with their operational concerns.

3.3. Development of a Software Prototype

The implementation of the system will ascertain the degree of interaction of business operators to make prudent business decisions for the needs of the retail facility. From the results of the prototype to generate relevant monitoring activity, an analysis will consider further research to improve the intelligence logic to suit the model of the simulation domain.

The needs of the retail organisation to make relevant decisions can affect the quality and marketability of inventory to potential customers. Retail employees need to recognise current marketing methodologies to improve underlying sales, by being in touch with customer needs. Records that are consistent with the buying patterns of inventory turnover are an important resource to understand the customer's buying preferences and the marketability of future stock.

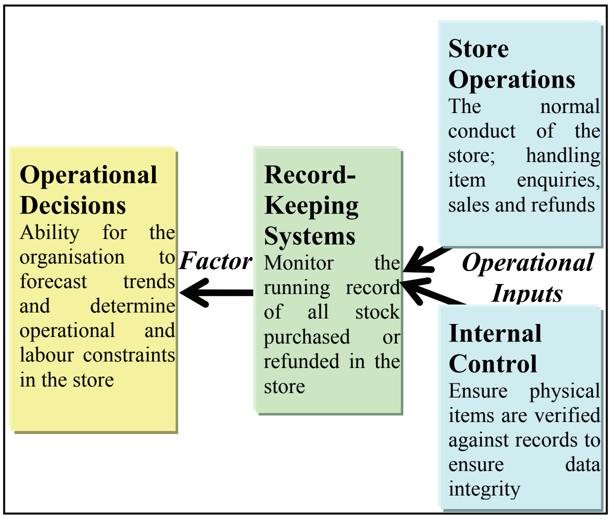

Store Operations
The normal conduct of the store; handling item enquiries, sales and refunds
Operational Decisions
Ability for the organisation to forecast trends and determine operational and labour constraints in the store
Factor
Record-Keeping Systems
Monitor the running record of all stock purchased or refunded in the store
Operational Inputs
Internal Control
Ensure physical items are verified against records to ensure data integrity

Figure 2: Operational Decision Chart (Blackwell 2003)

Computerised record-keeping systems are used in retail environments to obtain the necessary information to make operational decisions using the perpetual inventory accounting method. As shown in Figure 2, the record-keeping systems maintain a current record of the merchandise held by the store at any given time. This record is maintained by a computerised inquiry system, integrated into 'self-serve' kiosks or accessed by retail staff directly, to ensure that merchandise bought, replaced or refunded are accounted throughout the store's operation.

While computerised record-keeping management has modernised the process of retail transaction procedures, internal control is still necessary to ensure current inventory records of the store remain consistent. The annual or periodic stocktaking of a store is a core element of internal control to account for any discrepancies that may exist between the records and the physical merchandise held at any given time.

The context of the paper will examine the feasibility of using a multi-agent simulation to demonstrate the potential of improving internal control through the use of wireless sensor networks. Unlike traditional inventory control technologies such as bar-coding, wireless sensors with Radio Frequency Identification (RFID) technology can detect items that do not have to be read at line-of-sight. Thus, the manual process of stocktaking can be enhanced as long as the inventory is located within the proximity range of the wireless sensors.

4. DEVELOPMENT METHODOLOGY

The development of the Smart-Shop simulation system has been influenced in part by the industrial experience gained by the contributors of the paper, while permitting the ability for the contributors to reflect on the theoretical content of the material. The philosophy of all practice based engineering programs in University of Technology, Sydney (UTS) is reinforced with theory and industrial and commercial practice in the final year subjects Information & Communication Technology Analysis (ICTA) and Design (ICTD) (Chaczko et al 2006; Chaczko et al. 2004). The Australian Federal Government accredited program has been recognised as embodying world leading standards and practices, with educators all having held positions in industry as well as undertaking research programs; thus providing an additional academic quality process.

Out of the main design constraints when creating a new solution, foremost is that the system must be able to be completed and deployed within a predefined time. The constraint against an extensive prerequisite structure results in a limited number of core requirements, such that the basic technologies must be reviewed and assessed early in the development stage before important decisions are taken. The components of the software development processes are examined:

4.1. Software Technologies

The design of Smart-Shop System is based on the multi-agent simulation system by AnyLogic (Vangheluwe 2006). AnyLogic is a commercial application that generates the graphical agent-based models into Java code stubs. The code can be extended and customised for further development by importing the main libraries into a Java Development Environment such as Eclipse.

The main base object of the software agent forms the basis of which additional software agents are added together to suit the simulation model, which is made of Java objects instances in a vector array to achieve multi-agent capability. Interactivity between agents is achieved by establishing 'communication protocols' between agents, depending on the message content and information relayed, achieved by Remote Method

Invocation (RMI). State feedback and interactivity of the simulation environment is achieved through the 'root' object animating each instance of agent.

4.2. Software Quality Processes

Good quality assurance practice depends on the characteristic rules imposed upon the deliverables to ensure consistency and uniformity throughout the scope of the software development process (Barbacci 1996, Booch 1991).

- **Coding Structure:** Ensuring name conventions, the structure of each method and attribute declared, along with the file structure of the project remains consistent and modular.
- **Code Versioning:** Mechanisms to trace back software at all revision instances. This ensures that when the prototypes reach a level of stability, they will be branched off the development branch and backed up for archival purposes.
- **Document Versioning:** Versioning is implemented on all documentation produced, with the versioning rules based on incremental document revisions, according to minor/major document modification.

4.3. Software Analysis and Design

The formal analysis of the Smart-Shop System is categorised by the identification of major stakeholders in the system. This will assist in requirement analysis and ensure the completeness of software requirements.

The system is designed through an analysis into the stakeholders of the system and how they will interact with the simulation, of which to describe the environment of the Smart-Shop System within the scope of the users who will interact with the simulation. This will serve as a basis to involve the main stakeholders who will influence the final design implementation by encapsulating the system's composition and the interfaces for user interaction.

- **End-Users:** The business operators are the end users of the system. They will use and administer the software system for their modelling purposes, and as such there is an interest for the developers to ensure requirements fulfils their expectations.
- **Developers:** The contributors of this paper who are involved in the project and its maintenance. This includes the project manager to oversee project schedules, and the software engineers responsible for implementation.
- **Official Bodies:** Government and Retail bodies that supervise commercial practice standards and budgetary concerns. The stakeholder is in charge of enforcing legislation and/or organisational by-laws may affect the final product release.

4.4. Software Project Management

The concerns of the project quality, risk, and resources at each stage in the project lifecycle, are achieved by determining the software development process most suitable for the Smart-Shop System. This is examined through presenting the project results and artefacts.

The project was implemented according to a tailored prescription of the iterative development model. The rationale was the incremental nature of which the project was to be delivered. While the core elements of the project were complete, the project requirements had to be factored into the software implementation to satisfy the iterative prototyping philosophy in Figure 3.

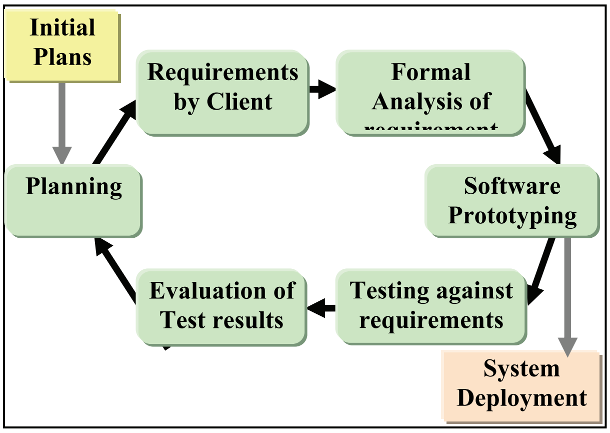

Initial Plans
Requirements by Client
Formal Analysis of requirement
Planning
Software Prototyping
Evaluation of Test results
Testing against requirements
System Deployment

Figure 3: Iterative Development Model (Kruchten 1996)

An important factor throughout the development strategy was to consider the change in scope as the implementation came into fruition, and thus ensure that the core requirements of the project continued to be satisfied with the relevant stakeholders of the Smart-Shop System. The simulation functionality was included incrementally as the prototype was demonstrated with the end-users of the system.

4.5. Software Architecture and Middleware

The role of software architecture and the selection criteria for software architecture had to be considered when developing the system prototype. An emphasis on implementing open infrastructure concepts and middleware was a core quality attribute for the system to ensure portability, scalability and interoperability of the Smart-Shop System. The system's software architecture is based on a distributed architecture. A distributed architecture would allow simulation agents to call a remote object, as it would when invoking another local object; in addition each software module can be deployed on different processors to allow processing load across multiple workstations.

The middleware implementation is achieved through the Jini Middleware System (Chaczko 2005), as its fundamental design allows for network 'plug-and-play' capability. While Jini itself is written in Java, the clients and services can be written as a wrapper around non-Java objects. The Jini System, known as a 'Federation', is a suite of clients and services communicating via the Jini Protocol, which implements the Java Remote Method Invocation (RMI) mechanism.

5. EFFECTIVENESS OF THE CONCEPT

An agent-based approach to software system analysis, design, simulation and delivery has significant benefits to the developers and end-users. In addition to this there are economic advantages that benefit the business community. These advantages have many dimensions that may pertain to effective use of ICT technology in inventory management strategies (O'Brien 1991; Yang 2003):

- Strategic Advantages: A net commercial benefit from improved management and awareness of critical projects are applications that fail less often, either perceived or real. Organisations are often not rewarded directly through higher reliability, scalability, robustness, maintainability and usability, yet these are tangible economic consequences from utilising better solutions.
- Knowledge Advantages: Developers will disseminate their skills and methods through their professional and informal contacts, thus providing a multiplier effect of the experience to the user.
- Commercial Applications: There is a strong potential for adaptation of similar to Smart-Shop System to venture into commercial applications beyond the retail domain.

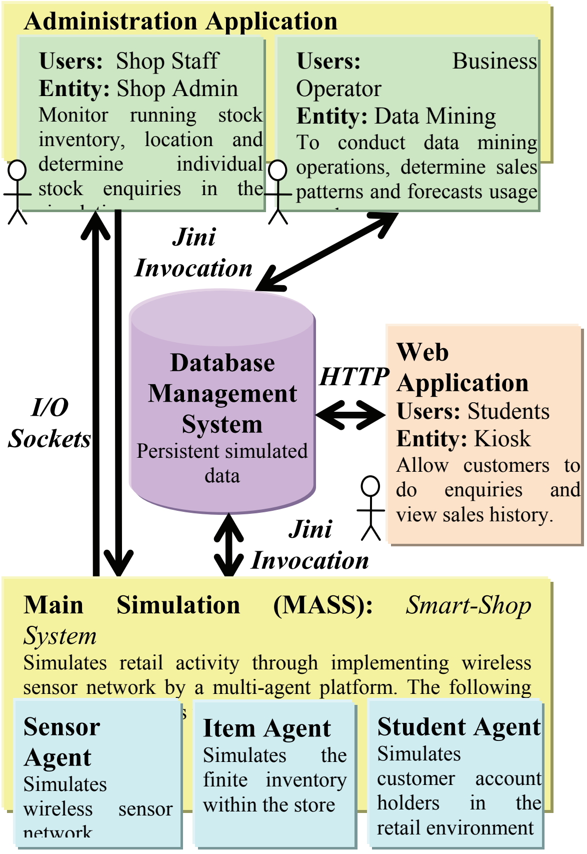

Administration Application
Users: Shop Staff
Entity: Shop Admin
Monitor running stock inventory, location and determine individual stock enquiries in the
Users: Business Operator
Entity: Data Mining
To conduct data mining operations, determine sales patterns and forecasts usage
Jini Invocation
I/O Sockets
Database Management System
Persistent simulated data
HTTP
Web Application
Users: Students
Entity: Kiosk
Allow customers to do enquiries and view sales history.
Jini Invocation
Main Simulation (MASS): Smart-Shop System
Simulates retail activity through implementing wireless sensor network by a multi-agent platform. The following
Sensor Agent
Simulates wireless sensor network
Item Agent
Simulates the finite inventory within the store
Student Agent
Simulates customer account holders in the retail environment

Figure 4: Smart-Shop System Architecture

As elaborated in Figure 4, the implementation comprises of a distributed architecture encompassing a multi-agent simulation that simulates the activity of a retail store that implements wireless sensors to track inventory. This simulation system will comprise of individual intelligent agents, including the store account holders, the merchandise and wireless sensors which will interact in a common shopping ground.

The design of the Smart-Shop System emphasises user customisation, in that scenarios are created to suit the unique layout of a store. Items can be placed at any given area, while pre-defined item attributes will affect the customer's behaviour and environment interactions, captured in visual depictions shown in Figure 5.

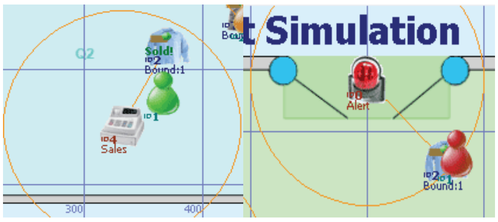


Figure 5: Simulated Sale and Attempted Theft by Agent

The Smart-Shop System in Figure 6 graphically displays transaction events, in which real-time activity including merchandise enquiries and stock-takes are handled by an administration application. Non real-time events, such as purchases or refunds and item tracking are saved to the database for data mining processes and store activity forecasting. Statistical graphs present the real-time state of the simulation in terms of quality of service, concerned with the wireless infrastructure's capability to handle transaction events in the store.

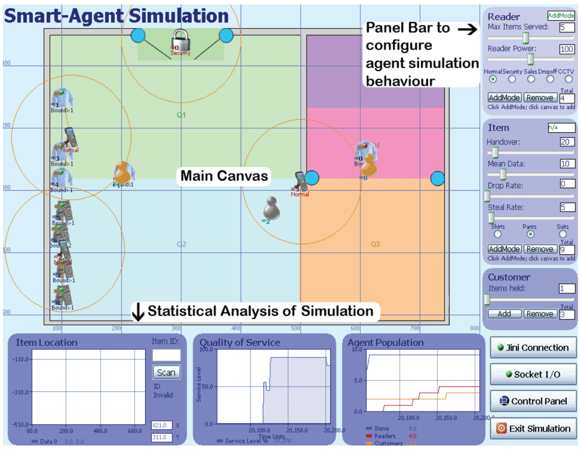


Figure 6: Smart-Shop System Application Screenshot

A preliminary run of the simulation to demonstrate the proof of concept is shown in Table 1 below. The simulation was run with the test case assumptions:

- Each individual scenario was run for approximately 2.4 hours, with the total test run elapsing after 12 hours.
- The simulation is constrained to 2 dimensions.

- The retail store area was made to 100m^2 to scale, split into 5 different rooms.
- 30 RFID readers were placed in an even order around the store, each with signal strength of 3m^2. This means that 90% of the store area is covered by the readers, with 10% non-signal coverage.
- 250 items affixed with tags will always be read the readers when they come into range, with environmental factors such as signal attenuation not factored into the simulation.
- 100 Customers are assumed to only have 4 distinct behaviours: to purchase items, steal items, acquire items for browsing, or return items.
- Items that are stolen will only be recorded by inventory. Aspects such as theft prevention through security guards and closed-circuit television are not factored into the simulation.
- Sales complete refers to total customer sales divided by total of all customers entering the store.
- Average Time Spent Shopping refers to the average time spent for each agent to browse for items
- Quality of Service refers to the percentage of number of inventory monitored compared to the total number of inventory in the store.

Table 1: Results from Test Case Proof-of-Concept

Scenario Micro-Perspective		Results: Macro-Perspectives		
Avg Item Value	Sales to Theft Ratio	Sales Completed	Average Time Spent	Quality of Service
$25	80:20	87.42%	5m 32s	72.15%
$50	70:30	73.81%	11m 12s	81.16%
$75	60:40	61.59%	15m 23s	85.71%
$100	50:50	49.82%	26m 3s	89.42%
$150	40:60	39.11%	31m 44s	88.91%

The simulation results can be interpreted according to the perspective of each user concern:

- Simulation micro-perspectives are of the concern of the test case executor. This involves collecting and entering statistical datasets of the retail environment to be modelled in the simulation. By setting the unique attribute of each stock, this will determine the customer's interaction with the stock.
- Simulation macro-perspectives deal with the concerns of the retail manager and decision makers. Quality of service data is highly significant to the end user, as the results will be interpreted to determine if a wireless sensor network setup is suitable, given their unique business scenarios.

The test scenarios results in Table 1 are only to be interpreted from a proof-of-concept perspective, given the artificial nature of the test constraints. The gathered results demonstrate that there is a potential basis for an improvement in the simulation model and input dataset, such that a more thorough analysis can be established to determine meaningful statistical trends.

6. CONCLUSION

The Smart-Shop System, through its design and implementation, achieves its principle objective of providing an accessible and effective software simulation for managing assets of retail organisations. By demonstrating the feasibility of implementing the system in an open distributed architecture with Jini, it accomplishes the integration of middleware infrastructure such that the end users can assess their inventory control strategies from a remote location.

A functional prototype of a retail environment is developed that incorporates wireless sensor network technology. The system simulates a generic sensor network system as an internal control mechanism, with the ability for end-users to examine their own circumstances for which sensor networks could improve their operational procedures. The use of statistical performance modelling allows the end user to quantify in real terms the benefits of using sensor network technology in a retail organisation. The simulated data can be examined through external data mining to plan placement strategies and forecast transaction trends.

ACKNOWLEDGEMENTS

Acknowledgements are made to the contribution of Michael Tran, Dr Perez Moses and Marina Capponi of Henry Bucks Menswear for their practical retail experiences and insights into the project.

REFERENCES

[1] IBM Corporation, 2006. *RFID Retail Case Study*, <http://ibm.com/industries/retail/doc/content/solution/1040471101.html>, Viewed 16 May 2006

[2] Blackwell G., 2003. *Wi-Fi Planet: RFID Review*, <http://www.wi-fiplanet.com/columns/article.php/3083401>, Viewed 12 May 2006

[3] Buckland M., 2004. *Redesigning Shop Services*, <http://sunsite.berkeley.edu/Literature/Shop/Redesigning/automatedlib.html>, Viewed 18 December 2006

[4] Vangheluwe H., 2002. *AnyLogic as a Agent Modelling Tool*, <http://moncs.cs.mcgill.ca/people/hv/teaching/MS/projects/sanghamitra/AnyLogic>, Viewed 11 December 2006

[5] Chaczko Z. et al, 2006. *7/24 Software Development in Virtual Student Exchange Groups: Redefining the Week*, ITHET '06, Sydney, Australia, July 2006

[6] Kruchten P., 1996. *A Rational Development Process*, Software Technology Support Centre Crosstalk Forum, July 1996

[7] Chaczko Z., D. Davis, V. Mahadevan, 2004. *New Perspectives on Teaching and Learning Software Systems Development in Large Groups.* 5th ITHET Conference 2004, p278

[8] Chaczko, Z, Davis J. D, Scott C., 2004. "*New Perspectives on Teaching and Learning Software Development in Large Groups- Telecollaboration*", IADIS International Conference WWW/Internet 2004 Madrid, Spain, 6-9 October 2004

[9] Chaczko Z, Marcel M, Lim L., 2005. *A middleware model for telecollaborative systems in education*, ITHET ‘05, Dominican Republic, July 2005.

[10] Barbacci, M. et al, 1996. *Principles for Evaluating the Quality Attributes of a Software Architecture*, Carnegie Mellon University, Software Engineering Institute, pp33-34

[11] Booch, G., 1991. *Object-Oriented Analysis and Design with Applications*, Addison-Wesley, p51

[12] Jacobson, I., 1998. *Object-Oriented Software Engineering: A Use Case Driven Approach: Revised Edition*, Addison-Wesley, pp57-58

[13] O'Brien, L. et al, 1991. *Retailing: Shopping, Society, Space, David Fulton Publishers*, London, pp71-75

[14] Yang, J. et al, 2003. *Smart Market Assistants & Recreation Trolley*, Beijing Institute of Technology, China, Beijing.

SUBPROBLEM SOLVING AND MACHINE PRIORITIZATION IN THE SHIFTING BOTTLENECK PROCEDURE FOR WEIGHTED TARDINESS JOB SHOPS

Roland Braune

Institute for Production and Logistics Management
Johannes Kepler University
Altenberger Straße 69
4040 Linz, Austria

roland.braune@jku.at

ABSTRACT

In this paper we perform investigations on the Shifting Bottleneck Procedure for weighted tardiness job shop scheduling problems. We propose machine prioritization rules which explicitly consider the specific structure of the occurring subproblems and compare them with conventional criteria. We study their effects in combination with alternative subproblem solution methods. Furthermore, we analyze the role of machine backtracking as an advanced control structure. Computational results are presented based on a set of adapted benchmark problems.

Keywords: job shop scheduling, total weighted tardiness, shifting bottleneck, machine prioritization

1. INTRODUCTION

Job Shop Scheduling (French 1982) involves sequencing a set of jobs on multiple machines. In the simplest case, each job is processed on each machine exactly once, whereby the processing orders (routings) are predetermined (precedence constraints) and can be different for each job. A job hence consists of several operations having fixed processing times assigned to them. A machine can only process one operation at a time (capacity / disjunctive constraint) and no preemption is allowed, i.e. once started, the processing of an operation cannot be interrupted until it has finished.

The most common optimization objective in job shop scheduling is certainly the minimization of the makespan, i.e. the maximum completion time $C_{\max}$ of all jobs.

However, other measures are at least that important and closer to the real world. Those include the number of tardy jobs, total completion or flow time and total weighted tardiness (TWT) $\sum w_j T_j$ which is subject of investigation in our contribution.

The tardiness T_j of a job j with respect to its due date d_j is computed as $T_j = \max(0, C_j - d_j)$, where C_j denotes the completion time of job j.

The associated problem may also include release times r_j for jobs and is denoted as

$$Jm \mid r_j \mid \sum w_j T_j$$

in the three-field notation of Graham, Lawler, Lenstra and Rinnooy Kan (1979).

Contrary to the makespan objective, weighted tardiness job shops have not attracted much attention in scheduling research, only a few contributions are available in this area. Among them are dedicated priority dispatch rules (Vepsalainen and Morton 1987; Anderson and Nyirenda 1990), local search based approaches (Kreipl 2000; de Bontridder 2005) and genetic algorithms (Mattfeld and Bierwirth 2004). This paper deals with an additional approach, the so called *Shifting Bottleneck Procedure* (SBP), initially proposed for the minimum makespan problem (Adams, Balas and Zawack 1988) and later adapted to $Jm \mid r_j \mid \sum w_j T_j$ by Pinedo and Singer (1999). We perform a computational study concerning the application of different subproblem solution procedures and machine prioritization rules within the shifting bottleneck procedure. Comparable studies have been published for $C_{\max}$ and $L_{\max}$ (maximum lateness) problems (Holtsclaw and Uzsoy 1996; Demirkol, Mehta and Uzsoy 1997; Aytug, Kempf and Uzsoy 2002). We extend these investigations to weighted tardiness job shops by considering the special characteristics of this kind of problem. Our paper is organized as follows:

In Section 2 we introduce the Shifting Bottleneck Procedure for job shop scheduling in general. Sections 3-6 describe its main tasks in the TWT context, with a special focus on machine prioritization and subproblem solution. Section 7 gives a brief overview on the concept of machine backtracking. In Section 8 we present our experimental results. A conclusion and an outlook on future research are given in Section 9.

2. SHIFTING BOTTLENECK PROCEDURES FOR JOB SHOP SCHEDULING

Shifting Bottleneck Procedures belong to the most popular optimization methods in the area of job shop scheduling. Established in the late 1980s by Adams, Balas and Zawack (1988), they have been successfully applied to a wide range of different problem setups both in theory and practice.

The main idea behind bottleneck scheduling is a decomposition of a job shop problem into several single machine problems. The single machine problems are solved separately, one after the other, always prioritizing the most critical machine with respect to a given bottleneck measure. Each single machine solution is inserted into a partial schedule for the enclosing job shop problem until all machines have been scheduled.

The procedure is based on the disjunctive graph representation of the underlying problem. During the progress of the SBP, the graph represents a partial solution to the job shop problem. Each time a subproblem is solved, the resulting sequence on the associated machine is inserted into the graph by orienting the (undirected) arcs between the operations on this machine. Let M be the set of all machines and M_0 the set of already scheduled machines. Then we can describe the flow of the shifting bottleneck procedure according to Figure 1.

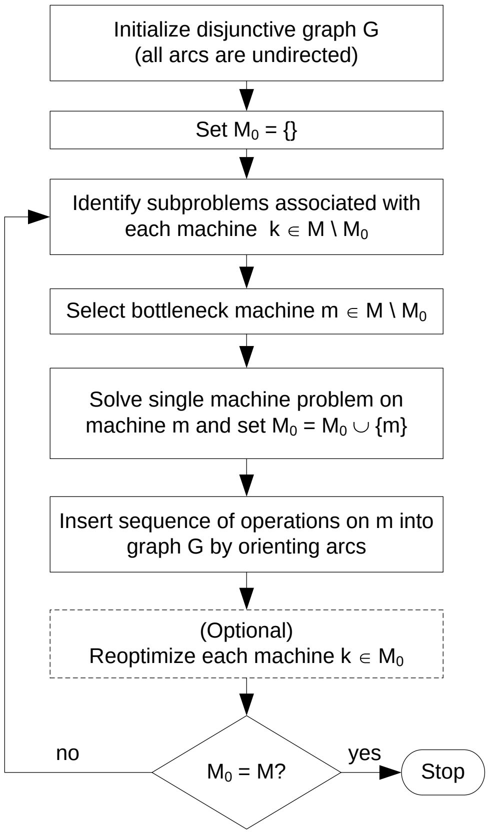

Initialize disjunctive graph G
(all arcs are undirected)
Set M₀ = {}
Identify subproblems associated with each machine k ∈ M \ M₀
Select bottleneck machine m ∈ M \ M₀
Solve single machine problem on machine m and set M₀ = M₀ ∪ {m}
Insert sequence of operations on m into graph G by orienting arcs
(Optional)
Reoptimize each machine k ∈ M₀
no
M₀ = M?
yes
Stop

Figure 1: Outline of the Shifting Bottleneck Procedure

Given this outline, the following four main tasks can be identified:

1. Subproblem identification
2. Bottleneck selection (machine prioritization)
3. Subproblem solution
4. Reoptimization

Furthermore, the SBP can be embedded into an enumeration framework in order to examine different machine orders and determine the best one (Adams, Balas and Zawack 1988).

Due to the fact that the graph structure of tardiness job shops differs considerably from those of makespan or maximum lateness problems, the tasks subproblem identification, bottleneck solution and subproblem solution require a dedicated approach as described by Pinedo and Singer (1999). In the following sections we give an overview on the tasks enumerated above in the context of tardiness job shop scheduling. Particularly the bottleneck selection and subproblem solution processes receive specific attention, as they represent the basis of our investigations.

3. SUBPROBLEM IDENTIFICATION

Consider a tardiness job shop with m machines and n jobs. In the graph representation of this job shop, there are n sink nodes, one for each job. When isolating a single machine scheduling problem (SMSP) from the graph, each operation in the SMSP has exactly n different due dates resulting from the longest paths between the respective operation and the sink nodes.

According to Pinedo and Singer (1999), the single machine problems can be described as follows: Let (i, j) be an operation to be scheduled on machine i, then d_{ij}^k denotes the (local) due date of operation (i, j) with respect to job k. Given the completion time C_{ij} of operation (i, j), the tardiness of the operation concerning job k can be computed as $T_{ij}^k = \max(C_{ij} - d_{ij}^k, 0)$. Since all jobs on machine i have to be scheduled, the actual tardiness of job k is determined as $\max_{(i,j)\in N_i} T_{ij}^k$, where N_i denotes the set of all operations on machine i. The total increase in the objective function given a schedule on machine i is

$$\sum_{k=1}^{n} w_k \left(\max_{(i,j)\in N_i} T_{ij}^k \right)$$

Considering delayed precedence constraints (DPCs) due to potential paths in the graph between operations on a machine, the single machine subproblems to be solved can be described as

$$1 \mid r_j, DPC \mid \sum_k w_k (\max_j T_j^k)$$

4. BOTTLENECK SELECTION

The selection of a bottleneck machine is an intrinsic step in the SBP. The definition of a bottleneck is not unambiguous however. As for the makespan case, there are many different ways of determining whether a machine should be prioritized over others (Holtsclaw and Uzsoy 1996; Aytug, Kempf and Uzsoy 2002). After a short introduction on the conventional quality based prioritization, we propose bottleneck selection criteria which are dedicated to the subproblem structure arising from tardiness job shops. The criteria are based on slack and problem infeasibility and need to be computed dynamically in each iteration of the SBP by analyzing the subproblems of all unscheduled machines.

4.1. Quality

The most common bottleneck selection criterion is oriented on the solution quality of the subproblems. In order to determine this measure, subproblems of all unscheduled machines have to be solved in advance and their resulting solution quality values are then ranked in descending order. The machine whose subproblem yields the highest TWT value is regarded as the bottleneck and its sequence gets inserted into the graph (Pinedo and Singer 1999).

4.2. Slack

The main idea behind this criterion is that the subproblem with the least slack with respect to its due dates may be regarded as the bottleneck. Due to the tighter due dates, it is more likely that the problem in question will increase the overall TWT. We distinguish two different types of slack: The slack of each operation with respect to its earliest local due date and the weighted minimum slack per job.

4.2.1. Earliest Due Date Slack ($SLCK$)

Consider an SMSP on machine i and let e_j be the earliest possible starting time of an operation j. Note that e_j is the maximum of the release time r_j and the earliest starting time of the operation due to potential delayed precedence constraints. Then for each operation j with processing time p_j a slack value can be computed as

$$slck_{ij} = \min_k(d_{ij}^k) - (e_j + p_j)$$

4.2.2. Weighted Minimum slack per job ($WSLCK$)

Contrary to $SLCK$, we compute the slack values not for each operation but for each job k of the superior TWT job shop problem. By this, we can include the job weights w_k into the calculation and therefore provide a more detailed estimate of the machine's criticality. The slack of an operation j regarding due date d_{ij}^k is defined as

$$slck_{ij}^k = \begin{cases} d_{ij}^k - (e_j + p_j) & \text{if } d_{ij}^k \text{ is defined} \\ \infty & \text{otherwise} \end{cases}$$

Since multiple operations may have a (local) due date regarding a job k, we are interested in the minimum slack value with respect to k. Additionally we want to consider job weights, hence we compute

$$wslck^k = 1/w_k \min_{(i,j)\in N_i}(slck_{ij}^k)$$

4.3. Infeasibility

The capacity (disjunctive) constraint stated in Section 1 enforces that no more than one operation at a time is processed on a machine. However, given time windows for operations defined by the release time r_j and a due date d_{ij}^k, it may occur that multiple operations require to be processed on a machine at the same time in order not to violate the due date(s). Such a situation can be detected by determining an infeasibility profile for the respective single machine problem (Aytug, Kempf and Uzsoy 2002).

We determine an infeasibility profile for the specific SMSPs described in Section 3 in the following way: For each operation j we define a time window ranging from the earliest starting time e_j to the earliest due date $\min_k(d_{ij}^k)$. The processing time p_j is distributed equally across the length of the time window, hence we obtain an average required capacity for any point within the time window.

By cumulating the time windows of all operations with their required capacities, we can create a capacity requirement profile for an SMSP as shown exemplarily in Figure 2.

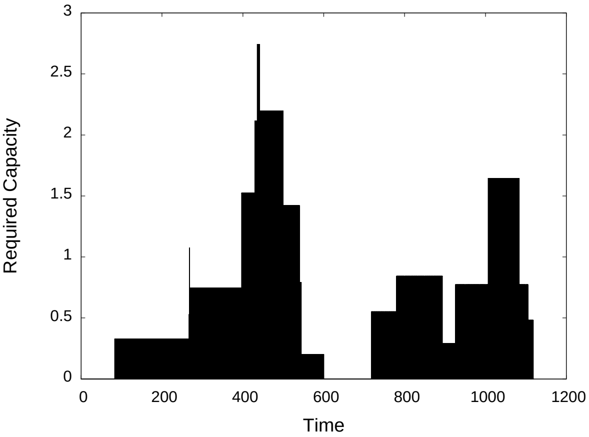


Figure 2: Example Infeasibility Profile

Such a profile can be referred to as an infeasibility profile in the context of disjunctive scheduling because cumulated average capacity requirements greater than 1 indicate a conflict situation. The machine is potentially required to process more than one operation at a time, which is infeasible due to the capacity constraint.

We use the infeasibility profile of a subproblem in order to determine two different bottleneck measures: The earliest due date infeasibility and the weighted average infeasibility per job.

4.3.1. Earliest due date infeasibility ($INFEAS$)

Given an infeasibility profile of an SMSP with respect to earliest due dates, we determine the area of the profile which exceeds the capacity limit. Let v_t be the (cumulated) capacity requirement at time t, which we refer to as the infeasibility value at time t. Based on this, we compute the earliest due date infeasibility bottleneck measure as

$$infeas = \sum_t \max(v_t - 1, 0)$$

4.3.2. Weighted avg. infeasibility / job ($WINFEAS$)

In order to take job weights into consideration we propose a further infeasibility related measure. For each job k of the superior tardiness job shop and each operation j we determine the time windows $[e_j, d_{ij}^k]$ and compute an average infeasibility value for each of these time windows:

$$infeas_{ij}^k = \begin{cases} (\sum_{t=e_j}^{d_{ij}^k} v_t)/(d_{ij}^k - e_j) & \text{if } d_{ij}^k \text{ is defined} \\ 0 & \text{otherwise} \end{cases}$$

We only keep the maximum value for each job k and multiply it with the associated weight w_k:

$$winfeas^k = w_k \max_{(i,j)\in N_i} (infeas_{ij}^k)^z$$

In order to amplify infeasibility values greater than one we actually square them ($z = 2$) before calculating the maximum.

5. SUBPROBLEM SOLVING

It is in every way important how well the occurring subproblems are solved: On the one hand, the solution quality may steer the bottleneck selection process itself as stated in Section 4.1. On the other hand, the resulting sequence on a bottleneck machine gets inserted into the overall partial solution and hence influences the whole subsequent optimization process.

Apart from solution quality, the computation time required for solving the SMSPs is an important factor in the context of bottleneck scheduling, especially when using machine backtracking and / or reoptimization.

Besides the beam search algorithm of Pinedo and Singer (1999), we use different local search methods for single machine optimization in our computational study. We apply a simple best improvement local search (BILS) procedure and a Tabu Search (TS) (Glover 1997) method. The approaches are conceptually similar to those presented in (Braune 2006). However, small modifications were necessary to respect the precedence constraints and the objective function of

$$1 \mid r_j, DPC \mid \sum_k w_k (\max{}_j T_j^k)$$

The local search methods rely on non-adjacent pairwise interchanges (NAPI) of jobs and use a modified ATC priority dispatch rule (Pinedo and Singer 1999) for generating the initial solution.

6. REOPTIMIZATION

Reoptimization may take place every time after a new single machine sequence has been inserted into the partial solution graph. According to the original idea proposed by Adams, Balas and Zawack (1988), a reoptimization cycle consists of the following steps:

1. Sort the set M_0 of already sequenced machines according to decreasing solution quality
2. For each machine $i \in M_0$ do the following:
   (a) Isolate the corresponding subproblem from the graph by removing all directed arcs between operations on machine i
   (b) Optimize the single machine problem and insert the resulting sequence into the graph
3. Unless the termination criterion is satisfied, goto step 1. Otherwise stop.

The reoptimization procedure terminates after a fixed number of reoptimization cycles or in case the most recent cycle did not yield a further improvement in overall solution quality. The latter principle is referred to as full reoptimization in the following.

7. BACKTRACKING

Backtracking or selective enumeration (Adams, Balas and Zawack 1988) can be used to examine various different machine sequences for a given problem. Each node in the search tree corresponds to a partial permutation of already scheduled machines. Branching is performed on the remaining unscheduled machines which are ranked according to the used bottleneck measure. The number of branches actually created at each level is controlled by an "aperture" parameter β (Pinedo and Singer 1999).

8. EXPERIMENTAL RESULTS

Our computational study is based on a set of 22 modified benchmark instances of size 10 x 10 taken from the OR-Library (Beasley 1990): ABZ5, ABZ6, LA16 – LA20, LA21 – LA24 (downsized by omitting the last 5 jobs), MT10 and ORB1 – ORB10. Originally intended for the makespan objective, these instances have been adapted by Pinedo and Singer (1999) in order to be able to use them for total weighted tardiness experiments. In fact they added a weight w_j and a due date d_j for each job. The due dates d_j were generated according to the following rule:

$$d_j = r_j + \left\lfloor f \cdot \sum_{i=1}^{10} p_{ij} \right\rfloor$$

where f denotes the tightness factor for due dates. We used the benchmark set with $f = 1.5$.

Since subproblem solution methods are also subject of our investigations, we first of all carried out a performance comparison on the single machine problem level. For this purpose, we sampled 30000 occurring SMSPs during the application of the SBP to the benchmark problem set. Optimal solutions are provided by the beam search method using the maximum aperture size. The solution quality obtained by the ATC priority dispatch rule (cf. Section 5) with $K = 2$ serves as a baseline. The local search methods (BILS, TS) have been parametrized according to Table 3.

The quality results are summarized in Table 1 in terms of mean percentage deviations from the optimal solution.

Table 1: Quality Deviations Obtained for Subproblems

	ATC	**BILS**	**TS**
Deviation	313,94 %	46,34 %	0,25%

Table 2: Computation Time for Subproblem Solution

	Beam Search	**BILS**	**TS**
Time (ms)	7953	250	239761

Table 3: Parametrization of LS Methods

	TS	**BILS**
Neighborhood	NAPI	NAPI
Initial solution	ATC	ATC
Neighborhood size	20	max
Tabu tenure	2	-
Max non-improving iterations	100	-

Furthermore, we compared the total running times of the four methods over all 30000 problems (cf. Table 2). Our investigations reveal that Tabu Search yields close to optimal solutions, but requires computation times which are orders of magnitude higher than for the (exact) Beam Search algorithm. Note that the running times of the ATC rule are negligibly small.

As a next step we applied the Shifting Bottleneck Procedure to the job shop benchmark set. We tested different combinations of bottleneck selection rules, subproblem solution methods and backtracking schemes. The resulting total weighted tardiness values are normalized using the weighted sum of job processing times (Lin, Goodman and Punch 1997):

$$(\sum w_j T_j)/(\sum w_j p_j)$$

The results are aggregated for the problem set, hence we report the sum of all individual TWT values.

For comparison purposes, Table 4 lists aggregated results from priority dispatch rules, Pinedo and Singer's SBP (SB-PS), their priority threshold backtracking heuristic (PTB) and their Branch and Bound algorithm which is able to solve the instances to optimality. Note that Singer and Pinedo imposed time limits for the applications of the SBPs, therefore results are not fully reproducible.

We run the SBP with and without backtracking (single pass) and always using full reoptimization. When backtracking is applied, we impose a limit on the number of solved SMSPs for each single benchmark instance. We believe that the computational effort can be better measured that way than in terms of CPU time. We compare the bottleneck selection criteria described in Section 4 with simple workload (total processing time) based and random prioritization. As for the subproblem solution methods, we apply Beam Search, BILS and Tabu Search when no backtracking is applied. Since TS is very time consuming we did not use it in the backtracking experiments.

The results obtained without backtracking, as summarized in Table 5, are moderate, yet most of them better than the PDR output. Quality and infeasibility based bottleneck selection rules obviously outperform the others, particularly the simple *INFEAS* rule performs very well.

Table 4: Results from PDRs and from Literature

Best PDR	**PTB**	**SB-PS ($\beta = 2$)**	**SB-PS ($\beta = 3$)**	**Opt.**
1,8189	1,3955	0,7426	0,6094	**0,5733**

Table 5: Single Pass Computational Results

Criterion	**Beam Search**	**BILS**	**TS**
quality	**1,3183**	1,6544	1,6013
SLCK (Σ)	1,7899	2,4005	1,7403
WSLCK (Σ)	1,6306	2,0908	1,7357
INFEAS	**1,1963**	1,6574	1,2730
WINFEAS (Σ)	**1,3809**	1,6904	1,4212
workload	1,6483	2,1408	1,7628
random	2,2178	2,2060	2,0005

Table 6: Results Obtained with Backtracking ($\beta = 2$) and a Limit of 100000 Solved SMSPs

Criterion	**Beam Search**	**BILS**
quality	0,6801	0,7910
SLCK (Σ)	0,6879	0,7808
WSLCK (Σ)	0,6864	0,7849
INFEAS	0,7284	0,8383
WINFEAS (Σ)	0,6973	0,7808
workload	0,7284	0,8839
random	0,7222	0,8392

Table 7: Results Obtained with Backtracking ($\beta = 3$) and a Limit of 300000 Solved SMSPs

Criterion	**Beam Search**	**BILS**
quality	**0,6237**	0,7059
SLCK (Σ)	0,6526	0,7429
WSLCK (Σ)	0,6578	0,7392
INFEAS	0,6519	0,7150
WINFEAS (Σ)	**0,6263**	0,7079
workload	0,6783	0,7741
random	0,6663	0,7801

As for the slack based measures, *WSLCK* exceeds *SLCK* which is possibly not meaningful enough to distinguish sharply between the machines. It further becomes clear that the choice of the subproblem solution approach significantly affects solution quality. While TS is only slightly worse than the Beam Search algorithm, the performance of the simple local search method considerably declines.

The incorporation of backtracking leads to a tremendous improvement in solution quality (cf. Table 6 and Table 7). In this context, it is striking that quality and *WINFEAS* perform almost equally well, with a slight though not significant advantage over the other rules. Hence we conjecture that *WINFEAS* is a good indicator for the expected solution quality. In general, it can be observed that the differences between the bottleneck selection rules are quite small. Even the random rule is not clearly inferior. We presume that the probability of generating good solutions increases with the number of examined machine sequences. To verify this conjecture, we enumerated all possible machine sequences for selected problem instances and discovered that the number of machine sequences yielding near-optimal solutions was very large. This observation coincides with the findings of Aytug, Kempf and Uzsoy (2002) for the makespan objective.

Again, the BILS subproblem solution method performs definitely worse compared to the exact Beam Search algorithm. It seems that the exact solution of the occurring subproblems is an essential factor in TWT oriented shifting bottleneck scheduling.

9. CONCLUSION AND OUTLOOK

We have presented a computational study of the Shifting Bottleneck Procedure for weighted tardiness job shop scheduling. On the one hand, we have developed dedicated bottleneck selection criteria and compared them to conventional ones. Empirical results show that the dedicated criteria yield consistently good and partly equivalent performance compared to the conventional quality based prioritization. The advantage of the proposed rules is mainly the fact that the subproblems need not be solved in advance in order to rank them. This may not be critical for small problems, but possibly a key factor for medium size or even large-scale instances.

On the other hand, we have analyzed the effect of applying alternative subproblem solution methods. Our computational results clearly indicate that solving the subproblems to optimality or at least close to optimality is vital in the TWT context. However, preliminary tests revealed that the running time of Pinedo and Singer's Beam Search algorithm increases dramatically with the problem size. As a consequence, trying to find optimal solutions this way can be considered infeasible for subproblem instances with more than 15 jobs. For this reason, we are convinced that efficient heuristic subproblem solving will play a central role when encountering larger instances.

Our future research will be directed towards an effective application of the SBP to medium-size and large-scale TWT job shops. According to preliminary experiments (Braune, Wagner and Affenzeller 2007), we assume that the required computation time renders backtracking impractical for large instances, even when using heuristic SMSP solvers. Therefore we intend to gain deeper insight into subproblem interaction in order to make the single pass procedure more competitive.

REFERENCES

Adams, J., Balas, E., Zawack, D., 1988. The shifting bottleneck procedure for job shop scheduling. *Management Science*, 34 (3), 391-401.

Anderson, E.J., Nyirenda, J.C., 1990. Two new rules to minimize tardiness in a job shop. *International Journal of Prod. Research*, 28 (12), 2277-2292.

Aytug, H., Kempf, K., Uzsoy, R., 2002. Measures of subproblem criticality in decomposition algorithms for shop scheduling. *International Journal of Production Research*, 41 (5), 865-882.

Beasley, J.E., 1990. *OR-Library*. Available from: http://people.brunel.ac.uk/~mastjjb/jeb/info.html [Accessed 17th July 2008].

De Bontridder, K.M.J., 2005. Minimizing total weighted tardiness in a generalized job shop. *Journal of Scheduling*, 8, 479-496.

Braune, R., Affenzeller, M., Wagner, S., 2006. Efficient heuristic optimization in single machine scheduling. *Proceedings of the International Mediterranean Modelling Multiconference I3M 2006*, 499-504. Barcelona.

Braune, R., Wagner, S., Affenzeller, M., 2007. Optimization methods for large-scale production scheduling problems. In: R. Moreno-Diaz et al., eds. *EUROCAST 2007, LNCS 4739*. Springer Verlag Heidelberg, 812-819.

Demirkol, E., Mehta, S., Uzsoy, R., 1997. A computational study of shifting bottleneck procedures for shop scheduling problems. *Journal of Heuristics*, 3, 111-137.

French, S., 1982. *Sequencing and Scheduling: An Introduction to the Mathematics of the Job Shop.* New York: Wiley.

Glover, F., 1997. *Tabu Search*. Kluwer Academic Publishers.

Graham, R.L., Lawler, E.L., Lenstra, J.K. and Rinnooy Kan, A.H.G., 1979. Optimization and approximation in deterministic sequencing and scheduling: a survey. *Annals of Operations Research,* 5, 187-326.

Holtsclaw, H.H., Uzsoy, R., 1996. Machine criticality measures and subproblem solution procedures in shifting bottleneck methods. *Journal of the Operational Research Society*, 47, 666-677.

Kreipl, S., 2000. A large step random walk for minimizing total weighted tardiness in a job shop. *Journal of Scheduling*, 3, 125-138.

Lin, S.-C., Goodman, E.D., Punch, W.F., 1997. A genetic algorithm approach to dynamic job shop scheduling problem. *Proceedings of the 7th International Conference on Genetic Algorithms 1997*, 481-488, East Lansing, USA.

Mattfeld, D.C., Bierwirth, C., 2004. An efficient genetic algorithm for job shop scheduling with tardiness objectives. *European Journal of Operational Research*, 155 (3), 616-630.

Pinedo, M., Singer, M., 1999. A shifting bottleneck heuristic for minimizing the total weighted tardiness in a job shop. *Naval Research Logistics*, 46 (1), 1-17.

Vepsalainen, P.J., Morton, T.E., 1987. Priority rules for job shops with weighted tardiness costs. *Management Science,* 33 (8), 1035-1047.

A CLUSTER-BASED OPTIMIZATION APPROACH FOR THE VEHICLE ROUTING PROBLEM WITH TIME WINDOWS

Michael Bögl

Institute for Production and Logistics Management
Johannes Kepler University
Altenberger Straße 69
4040 Linz, Austria

michael.boegl@jku.at

ABSTRACT

In this paper we describe a simple cluster approach for solving the Vehicle Routing Problem with Time Windows (VRPTW). The idea is to combine heuristic and exact approaches to solve the problem. This is done by decomposing the problem into two sub-problems. The first step consists of building customer clusters, in the second step, those clusters are solved with an exact method. We present computational results for the Solomon benchmark problems.

Keywords: vehicle routing, time windows, VRPTW, metaheuristic, tabu search, multi-objective optimization

1. INTRODUCTION

Transportation problems arise in a lot of companies. In a typical supply chain these are the sourcing of raw materials, the in-company distribution, the distribution to the retailers and reverse logistics. Furthermore it is apparent in the supply of services like waste collection, public transport, mail distribution and others. Transportation not only causes a lot of cost but also pollutes the environment. The pressure of the market forces firms to cut cost down. Computerized planning of the distribution process have shown to produce considerable savings of about 5% to 20% (depending on the application) (Toth and Vigo 2001).

Research for the Vehicle Routing Problem (VRP) was started nearly 50 years ago by Dantzig and Ramser (1959). They tried to solve a route planning problem for gasoline distribution, where a set N of n station points (or customers respectively nodes) $p_1, p_2, ..., p_n$ are given and deliveries to them are made from node p_0, called terminal point or depot. A delivery vector is given, it specifies the demand q_i for every node i. A fleet K of k trucks is available; each truck has a capacity Q, where $Q > \max q_i$, i.e. the demand of every node is smaller than the truck capacity. And there is an arc set A, where $c_{ij} \in A$ are the travel cost from node p_i to p_j. They developed the first Linear Programming (LP) based approach for the VRP. At that time Integer Linear Programming (ILP) was in the beginning of its development, as noted in Dantzig and Ramser (1959).

The next milestone was set by Clarke and Wright (1964) with their savings heuristic. Their approach is simple and fast, which explains its popularity even nowadays (Laporte and Semet 2001). Many solution concepts for more complex problems (with a lot of different constraints) are based on that heuristic or use it to compute an initial solution.

Those two methods present two main research directions which were taken the last decades: exact versus heuristic approaches. A long time those two approaches were not combined. This has different causes. Among other things, by using heuristic approaches one spoils the chance to find the global optimum. On the other hand: exact approaches may take an inconsiderable long computation time (to find the optimum). For practitioners there is another important reason: most of the time it is not necessary to find the best solution, a solution which is better than current practice is often enough (Wren 1998).

From the first appearance of the VRP until now a lot of different generalizations evolved. Among others there are the VRP with multiple depots (MDVRP, MVRP), the VRP with stochastic demands (SDVRP, SVRP), Pickup and Delivery VRP (PDVRP), VRP with Backhauls (VRPB), or the VRP with Time Windows (VRPTW) (Dorronsoro 2007; Toth and Vigo 2001).

This work focuses on the VRPTW. It is NP-hard (cf. Cordeau et al. 2001) and consists of the following constraints: every node has a time window, i.e. a node p_i may only be serviced between its earliest arrival time (a_i) and latest arrival time (b_i). Every stop at a node takes a certain service time (s_i). If a vehicle arrives before a_i it must wait until then to begin service. After servicing the node the vehicle may leave and continue to service the next node or return to the depot. Every vehicle must return to the depot before a certain time, i.e. every route has a maximal time. All other constraints are the same as in the VRP (Cordeau et al. 2001). Before we can give a formal description of the VRPTW we need to define the set $V = N \cup \{p_0\}$ and the following variables: x_{ijk} define the flow of the vehicles, i.e. x_{ijk}, $(i,j) \in A, k \in K$ is equal to 1 if arc (i,j) is serviced by vehicle *k*, 0 otherwise. Time variables w_{ik} define the time of service of vehicle *k* at node *i*.

Now the VRPTW can be formally described:

$$\min \sum_{k \in K} \sum_{(i,j) \in A} c_{ij} x_{ijk} \tag{1}$$

subject to

$$\sum_{k \in K} \sum_{j \in V} x_{ijk} = 1 \quad \forall i \in N \tag{2}$$

$$\sum_{j \in N} x_{0jk} = 1 \quad \forall k \in K \tag{3}$$

$$\sum_{i \in V} x_{ihk} - \sum_{\substack{j \in V \\ i \neq j}} x_{hjk} = 0 \quad \forall h \in N, \forall k \in K \tag{4}$$

$$\sum_{i \in N} x_{i0k} = 1 \quad \forall k \in K \tag{5}$$

$$\sum_{i \in N} q_i \sum_{j \in V} x_{ijk} \leq Q \quad \forall k \in K \tag{6}$$

$$w_{ik} + s_i + c_{ij} - M(1 - x_{ijk}) \leq w_{jk} \quad \forall i, j \in V, i \neq j, \forall k \in K \tag{7}$$

$$a_i \leq w_{ik} \leq b_i \quad \forall i \in V, \forall k \in K \tag{8}$$

$$x_{ijk} \in \{0,1\} \quad \forall (i,j) \in A, \forall k \in K \tag{9}$$

$$w_{ik} \geq 0 \quad \forall i \in N, \forall k \in K \tag{10}$$

The objective function (1) minimizes the length of the routes. Constraint (2) ensures that every node is visited exactly once. Constraints (3), (4) and (5) force every vehicle to leave the depot, move from one node to the next and finish the tour at the depot. Constraint (6) ensures that the capacity of every vehicle is not exceeded. Constraints (7) and (8) guarantee feasible schedules with respect to time windows. Constraint (9) enforces binary decision variables for the arcs. Finally, constraint (10) forces time variables to be positive. Please note: in this model the travel time equals the travel cost.

In this formal description the objective function minimizes the total route length. This may not always be the main goal. Another often used objective function is to minimize the vehicle fleet size and the total length of the routes. Then the VRPTW is a multi-objective optimization problem. But there are other goals one might want to achieve, e.g. the minimization of the waiting time (e.g. transportation of passengers) or the minimization of the total routing time (e.g. transportation of hazardous materials). In this work our primary optimization goal is the minimization of vehicles/routes and the secondary optimization goal is the minimization of the total length of the routes.

We will continue by giving a short literature review in section 2; section 3 motivates the research direction which this work is aimed at. Section 4 describes our solution method; section 5 shows results of computational experiments and at last section 6 gives an outlook on further ideas, refinements and research directions.

2. LITERATURE REVIEW

The VRPTW was subject of intensive research. Hence numerous different solution concepts were developed. The spectrum reaches from simple one stage heuristics (Solomon 1987), two stage heuristics (cluster-first, route-second; route-first, cluster-second) (Dorronsoro 2007; Cordeau et al. 2001), various local search methods (Bräysy, Hasle and Dullaert 2004), metaheuristics (Thangiah 1995; Potvin et al. 1996; Homberger and Gehring 2005) to exact approaches based on mathematical programming (Kallehauge, Larsen and Madsen 2001).

Solomon (1987) introduced three heuristics (I1, I2 and I3). They differ mainly in the weighting function. The I1 heuristic is nowadays the most popular. It is a sequential route construction heuristic and inserts customers into the current route according to a weighting function. It determines the best insertion position for all unrouted customers in the actual route by calculating the weighted sum of the detour and the shift of arrival time of the following customers. The customer for which the difference between the distance to the depot and the weighted sum of the best insertion position is maximized is inserted. This is repeated until all customers are serviced. If no customer can be inserted a new route is started.

A clustering metaheuristic was proposed by Thangiah (1995). He developed a system called GIDEON. The customers are sorted according to their polar coordinates (the depot is origin) and clustered into sectors. An insertion heuristic creates routes. The clusters are then optimized using a genetic algorithm.

One of the most successful metaheuristic approaches for large problem instances was developed by Homberger and Gehring (2005). It is a two phase metaheuristic. In the first phase an Evolution Strategy minimizes the number of routes, in the second phase a Tabu Search metaheuristic minimizes the total length of the routes.

3. MOTIVATION FOR A CLUSTER-BASED APPROACH

Even though instances up to 1000 nodes can be solved to optimality, benchmark problems with 100 customers are not yet solved to optimality (cf. Kallehauge, Larsen and Madsen 2001).

Only problems up to a few dozen of nodes can easily be solved to optimum nowadays (cf. Toth and Vigo 2001). For bigger instances (50 or more nodes) one possibility would be to decompose the problem into smaller sub-problems, which then can be solved faster by exact methods.

We think that sophisticated clustering methods for the VRPTW would contribute to the solution process of large scale problems. Because the problem size decreases, hence the effort to calculate the exact solution for a single cluster decreases too. Additionally, as nowadays the trend goes towards parallel computing (multi-core personal computers, GRID computing) clustering algorithms may contribute to the parallelization of solution concepts.

Here, an optimal clustering is a set of clusters, where every cluster contains the customers of the optimal routes. Due to the presence of time windows complex clusters arise. Figure 1 shows a solution for a 50 node benchmark problem (instance R101 of the Solomon (1987) benchmark problems; a short description is in section 5). The black rectangle in the middle is the depot; the Bezier curve in the upper left corner marks a simple cluster and the Bezier curve next to the depot marks a more complex cluster. Optimal solutions for bigger problems have a similar clustering (cf. Kallehauge, Larsen and Madsen (2001) for the optimal solution of a 100 customer problem).

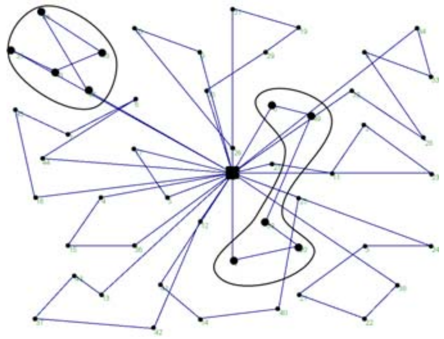


Figure 1: Solution of a 50 Nodes VRPTW Benchmark Problem of the Solomon (1987) Benchmarks

Recently a few researchers developed cluster algorithms for vehicle-routing like problems, e.g. Dondo and Cerdá (2007) propose a cluster based optimization approach for the multi-depot VRPTW.

4. SOLUTION CONCEPT

As mentioned earlier, we developed a cluster based optimization approach, i.e. our focus is to find clusters, from which optimal solutions can be derived. What we do is group customers into clusters and then calculate the route for every cluster. We start with an initial clustering and then iteratively change the customer clusters.

The general flow of the algorithm is depicted in figure 2. At first an initial clustering is calculated. Then the routing problem is solved for every cluster and the quality of the solution is evaluated. The variables *actual_solution* and *best_solution* are initialized with this solution. Now the main optimization loop is entered. A probabilistic Tabu Search metaheuristic optimizes the clusters using different neighborhood operators until a termination criterion is met.

Every solution cycle generates a certain predefined amount of neighbors probabilistically. The quality of these candidate solutions is evaluated and the best non-tabu solution is selected as the new *actual_solution* for the next iteration. If *actual_solution* is better than *best_solution*, *actual_solution* is assigned to *best_solution*. Basically, this is a standard Tabu Search algorithm.

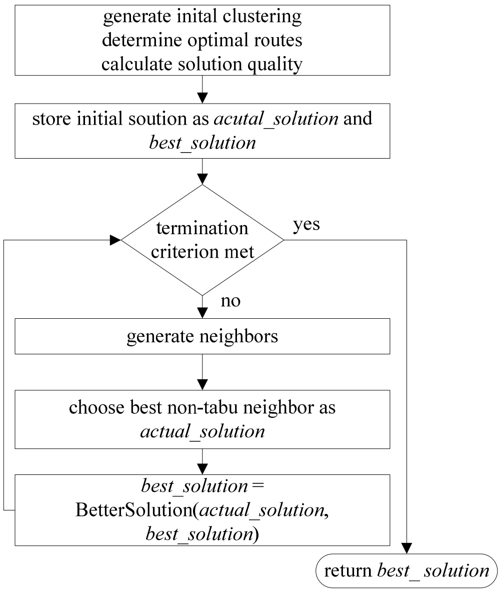

generate inital clustering
determine optimal routes
calculate solution quality
store initial soution as *acutal_solution* and *best_solution*
termination criterion met
yes
no
generate neighbors
choose best non-tabu neighbor as *actual_solution*
best_solution = BetterSolution(*actual_solution*, *best_solution*)
return *best_ solution*

Figure 2: Tabu Search Flow Diagram

4.1. Solution Evaluation

A neighbor respectively a solution in this context is a set of clusters, where a cluster is again a set of customers. A cluster consists of customers who can feasibly be serviced by one vehicle, i.e. every cluster represents one tour.

We somehow must determine the quality of such a cluster. The following possibilities exist:

1. Use a lower bound method
2. Use an upper bound method
3. Use an exact method

Method 1 is fast, but e.g. bounds based on integer relaxation for the VRPTW are rather weak (cf. Cordeau et al. 2001) and we risk infeasible clusters, i.e. clusters for which no feasible route exists. Another fast approach would be to use an upper bound method. Even though one always gets a feasible solution, the disadvantage is that one can not say how good the solution is.

We decided to use an exact method and solve the routing problem to the global optimum. We use the MILP formulation given in section 1 and ILOG CPLEX 11 to solve it.

The disadvantage of this approach is that the execution time greatly depends on the problem instance. For example in Kallehauge, Larsen and Madsen (2001) the execution time for the 25 node problems of the Solomon (1987) benchmark problems varies between

0.1 second (instance R101) and 221087.1 seconds (instance RC207).

Because the calculation of the optimal solution can take a considerable amount of time, we cache results of solved routing problems in a so called cluster table. Each time a cluster is generated, it is looked up there. If it was already encountered, a table entry exists and the according value is taken. If no entry in the table exists, the MILP solver is called, the routing problem is solved and the result (length of the route) is stored in the cluster table.

This has the advantage that if we encounter a cluster again later in the search process, it is not necessary to optimize it again. Additionally we can discard special clusters, which we have not seen yet. Let's consider two clusters (i.e. two sets of customers) $y_1 = \{p_1, p_2, \ldots, p_m\}$ and $y_2 = \{p_1, p_2, \ldots, p_m, \ldots, p_n\}$, where $y_1 \subseteq y_2$. Then, if y_1 is infeasible (i.e. no route exists which satisfies all constraints), y_2 must be infeasible too.

A solution to the routing problem can take two values: either the length of the route or infeasible.

As the VRPTW here is seen as multi-objective optimization problem, it is necessary to keep in mind that our primary optimization goal is the minimization of the number of used vehicles (i.e. number of routes). The secondary optimization goal is the minimization of the length of the routes.

Objective function values are sorted lexicographically, i.e. a solution with two routes is always better than a solution with three routes regardless of the tour length.

4.2. Initial Clustering

Before we can start the main search process we need an initial clustering. We implemented two methods.

The first initialization method is based on the Solomon I1 heuristic. We partition the customers into clusters according to the routes generated by the I1 heuristic. For example if we solve 25 customer problem C104 with the Solomon algorithm we may get, depending on the parameters, the following three routes (0 is the depot, numbers 1 to 25 are the nodes; routes are written in a permutation like style, i.e. (0, 2, 5, 0) means the vehicle starts at node 0 (depot), visits node 2 then node 5 and finally returns to the depot): {(0, 13, 15, 17, 18, 19, 16, 14, 12, 0), (0, 20, 24, 25, 22, 11, 9, 23, 21, 10, 6, 2, 0), (0, 2, 7, 8, 4, 1, 3, 5, 0)} with a total length of 254.88. We now use these three customer sets as clusters and solve the according routing problem using a MILP solver. Every route is solved separately, i.e. in this example the MILP solver is called three times. The following three routes are returned: {(0, 13, 17, 18, 19, 15, 16, 14, 12, 0), (0, 24, 25, 2, 6, 11, 9, 10, 23, 22, 21, 20, 0), (0, 2, 7, 8, 4, 1, 3, 5, 0)} with a length of 233.65. Our initial solution has the quality (3, 233.65).

The second method we implemented is called one-node-per-cluster. As the name suggests, every customer builds a separate cluster. It is a very naïve method. The idea in using such a simple initialization is that if we encounter a lot of small infeasible clusters in the beginning of the search, the algorithm has gathered more knowledge about the problem and can discard more infeasible clusters later.

4.3. Neighborhood Operators

Neighborhood operators are used to produce new solutions and a crucial element of the Tabu Search metaheuristic (Gendreau and Potvin 2005).

Here the generation of a neighbor consists of two steps. At first the clusters are modified. The second step is the calculation of the optimal routes for the changed clusters. The length of the optimal route reflects the quality of the cluster.

Two different neighborhood operators were implemented: exchange and move. Those two types are typical operators used in the context of vehicle routing (cf. Homberger and Gehring 2005).

The move neighborhood operator takes the actual solution, selects two random clusters, e.g. $c_1 = \{p_a, p_b, p_c, \ldots, r, \ldots, p_k\}$ and $c_2 = \{p_n, p_o, p_q, p_r, \ldots, p_u\}$ and moves a random, non-forbidden node r from c_1 to c_2.

Now both new clusters are looked up in the cluster table. If it already contains an entry for both or either of the new clusters, the according quality value is taken. Missing values are calculated using the MILP solver.

If c_2 is infeasible the move is reversed, node r is temporary forbidden and another node of set c_1 is chosen. This step is repeated until a feasible solution is found or all nodes of c_1 are forbidden during this neighborhood operation. All encountered trial solutions are stored in the cluster table.

The move operator is able to decrease the number of routes in contrast to the exchange operator.

The exchange operator selects two random clusters $c_1 = \{p_a, p_b, \ldots, k, \ldots, p_i\}$ and $c_2 = \{p_s, p_t, \ldots, r, \ldots, p_u\}$. From cluster c_1 a random, non-forbidden node k is chosen, from cluster c_2 a random, non-forbidden node r is chosen. Those two nodes are exchanged, and yield to the following clusters: $c_1 = \{p_a, p_b, \ldots, p_i, r\}$, $c_2 = \{p_s, p_t, \ldots, p_u, k\}$. Again, both new clusters are looked up in the cluster table and in case of a missing value, the MILP solver is called. In case any of the clusters is infeasible the move is reversed and nodes k and r are not allowed to be exchanged during this neighborhood operation.

4.4. Tabu List

As the Tabu Search metaheuristic is a neighborhood based steepest descent search method it must somehow be ensured that the search process can escape from local optima. This is the purpose of the tabu list (or memory).

Generally, tabu lists are used to control the search process, i.e. short term memory may be used to prevent from cycling and intermediate and long term memory may be used to intensify or diversify the search process (Gendreau and Potvin 2005).

Here, we use short term memory to prevent cycling. No long term or intermediate memory is used.

We do not only store solution attributes but complete solutions. Those solutions must not be selected again.

Now the description of the developed solution concept is complete. In summary, it is a probabilistic Tabu Search with short term memory and two different neighborhood operators.

5. COMPUTATIONAL RESULTS

All computational experiments were done using the Solomon (1987) benchmark instances. They consist of different classes of problems: R, C and RC. Class R contains random node positions. The nodes in class C are clustered. Class RC contains a mixture of class R and class C. Every problem class contains two different sets of problem instances: 1 and 2. Class 1 contains problem instances with narrow time windows and class 2 contains problem instances with wide time windows. There are 6 different problem classes. R1, R2, C1, C2, RC1, RC2. Every problem class contains between 8 and 12 problem instances. Solomon (1987) created problem instances with 25, 50 and 100 customers. Distances are Euclidean and calculated using node coordinates.

We present 4 different test-runs. Each was run twice using another random seed. Table 1 shows the configuration of the 4 test-runs. A test-run was stopped after one of the termination criterions was met: either the iteration counter reached the iteration limit or execution time exceeded the maximal execution time. If more than one neighborhood operator is given, one is chosen randomly for every neighborhood generation.

Table 1: Configuration of the 4 Test Runs

	Test #1	Test #2	Test #3	Test #4
Iterations	1000	1000	1000	1000
Max. Execution Time	15 min.	15 min.	15 min.	15 min.
Neighborhood Size	20	20	20	20
Tabulist Length	1000	1000	1000	1000
Initialization	one-node-per-cluster	Solomon I1	one-node-per-cluster	Solomon I1
Neighborhood Operator	move	move	move, exchange	move, exchange

Table 2: Averaged (2 Runs) Solution Quality for all 25-Node-Solomon Instances

Label	Test #1		Test #2		Test #3		Test #4	
	V.	Length	V.	Length	V.	Length	V.	Length
C101	*3*	*191,81*	*3*	*191,81*	*3*	*191,81*	*3*	215,57
C102	*3*	*190,74*	*3*	242,15	*3*	*190,74*	*3*	242,15
C103	*3*	203,62	*3*	*190,74*	*3*	208,46	*3*	202,40
C104	*3*	221,53	*3*	203,63	4	256,50	*3*	212,75
C105	*3*	*191,81*	*3*	*191,81*	*3*	*191,81*	*3*	*191,81*
C106	*3*	*191,81*	*3*	*191,81*	*3*	*191,81*	3,5	223,55
C107	*3*	*191,81*	*3*	*191,81*	*3*	*191,81*	*3*	*191,81*
C108	*3*	*191,81*	*3*	*191,81*	*3*	*191,81*	*3*	*191,81*
C109	*3*	*191,81*	*3*	213,07	*3*	*191,81*	*3*	225,85
C201	*2*	*215,54*	*2*	*215,54*	*2*	*215,54*	*2*	*215,54*
C202	*2*	*215,54*	*1*	*223,31*	*1,5*	*219,43*	*1*	*223,31*
C203	*2*	*215,54*	*1,5*	*241,14*	*2*	*215,54*	*1*	*224,46*
C204	2,5	235,50	*1*	*213,93*	2,5	240,19	*1*	214,35
C205	*1,5*	*256,50*	*1*	*297,45*	*1,5*	*256,50*	*1*	*297,45*
C206	*1,5*	*250,47*	*1*	*287,21*	*2*	*215,54*	*1*	*288,02*
C207	*2*	*215,34*	*1*	*274,78*	2,5	238,98	*1*	*274,78*
C208	*2*	*215,37*	*1*	*229,84*	*2*	*215,37*	*1*	*229,84*
R101	*8*	*618,33*	*8*	*618,33*	*8*	626,77	8,5	670,60
R102	*7*	558,12	*7*	*548,11*	*7*	*548,53*	*7*	*548,53*
R103	*4*	*473,39*	*4*	*495,02*	5	473,19	*4*	*490,56*
R104	*4*	425,18	5	498,16	*4*	424,53	4,5	463,89
R105	*5*	*559,84*	*5,5*	*547,35*	*5,5*	*544,13*	*5,5*	*558,90*
R106	*5*	*466,48*	*5*	*466,48*	*5*	*467,72*	*5*	*466,48*
R107	*4*	447,75	*4*	451,42	*4*	446,66	4,5	456,24
R108	*4*	*398,29*	*4*	442,72	*4*	406,47	*4*	443,43
R109	*4*	*467,28*	*4*	*460,52*	7	538,58	*4*	*460,52*
R110	*4*	*449,41*	*4*	*454,42*	*4*	*451,14*	*4*	*464,54*
R111	*4*	436,06	*4*	434,18	*4*	435,49	*4*	449,35
R112	*4*	404,15	*4*	404,10	*4*	407,58	*4*	406,40
R201	*2*	*525,25*	*2*	*525,25*	*2,5*	*502,80*	*2*	*528,93*
R202	*3,5*	*415,04*	*2*	*529,57*	*3,5*	*421,13*	*2*	*528,80*
R203	*3*	416,57	*2*	*447,01*	*3*	404,59	*2*	*474,28*
R204	*2*	*360,79*	*1*	*483,59*	*2,5*	*386,64*	*1*	*479,65*
R205	*2*	*412,81*	*2*	*405,98*	*2*	*419,24*	*2*	*419,20*
R206	*2*	*389,43*	*2*	*395,69*	*2*	*380,76*	*2*	*404,39*
R207	*2*	*372,45*	*1*	*408,60*	*2*	*374,32*	*1*	*408,60*
R208	2	335,69	*1*	*329,33*	2	336,07	*1*	*329,33*
R209	2,5	375,24	*2*	383,24	2,5	376,65	*2*	396,49
R210	*3*	411,61	*2*	*414,18*	*3*	422,58	*2*	*420,74*
R211	3	362,63	*2*	426,20	3	371,21	*2*	419,40
RC101	*4*	463,60	*4*	463,60	4,5	469,56	4,5	470,84
RC102	*3*	*352,74*	*3*	*352,74*	*3*	*352,74*	*3*	*352,74*
RC103	*3*	*333,92*	*3*	*333,92*	*3*	*333,92*	*3*	*333,92*
RC104	*3*	*307,14*	*3*	*307,14*	4	367,10	*3*	*307,14*
RC105	*4*	*412,38*	*4*	*412,38*	*4*	*412,38*	*4*	*412,38*
RC106	*3*	*346,51*	*3*	*346,51*	*3*	*346,51*	*3*	*346,51*
RC107	*3*	*298,95*	*3*	*298,95*	3,5	329,00	*3*	*298,95*
RC108	*3*	*294,99*	*3*	*294,99*	4	394,86	*3*	*294,99*
RC201	*2*	*432,30*	*2*	*450,43*	*2*	*433,26*	*2*	*468,56*
RC202	*3*	*338,82*	-	-	*3*	355,28	-	-
RC203	3	362,34	*1*	*521,51*	3,5	*357,09*	*1*	*506,84*
RC204	*3*	*300,23*	*1*	*344,01*	*3*	*355,04*	*1*	*342,70*
RC205	*3*	*338,93*	*2*	*512,49*	*2,5*	*391,75*	*2*	*529,22*
RC206	*3*	*325,10*	*2*	*450,84*	*2*	*361,47*	*2*	*473,13*
RC207	*3*	314,74	*2*	*427,36*	*3*	330,67	*2*	*474,82*
RC208	*3*	*312,71*	*2*	*393,47*	*4*	*396,23*	*2*	*458,37*

Computational experiments were done on an Intel Pentium 4 Mobile with 768 MB RAM. The solution concept was implemented in C# (.NET Framework 3.5). We used ILOG CPLEX 11 to solve the MILP sub-problem using the model presented in section 1.

Optimal solutions were taken from Kallehauge, Larsen and Madsen (2001). They cut off distances after the second decimal place. We use double precision. Their objective was the minimization of the tour length. They did not publish results for the following instances: R204, RC204 and RC208.

Table 2 shows the averaged results over two test-runs for all 25 customer problems of Solomon (1987). For every test-run, the column labeled V. contains the number of used vehicles (nr. of routes); the column labeled Length contains the total length of the solution. Italic entries mark solutions, which are closer than 0.5% to the given optimal quality (because of the difference in distance). If our solution to a problem instance uses fewer vehicles than the optimal solution, then the value is underlined (and both values are written in italic).

In columns Test #2 and Test #3 results for problem instance RC202 are missing, because the computer run out of memory when calculating the optimal routes for the initial solution.

Results for Test #1 show that our approach finds a solution which is at most 0.5% worse than the reference solution in 40 cases out of 56. In column Test #2 even 44 problems are within the 0.5% gap. In Test #3 we can solve 33 problems close to the optimum and in Test #4 39 problems. the Solomon I1 heuristic clustering seems to be better than the one-node-per-cluster initialization. Results for Test #3 and #4 (exchange and move operator randomly chosen) are inferior compared to Test #1 and #2 (only move operator).

6. CONCLUSIONS

Or approach is able to find near optimal solutions within reasonable time for small problems.

The main drawbacks of the developed solution concept are the lack of an appropriate diversification method and the weak MILP formulation for the VRPTW. The next step will be the development of an appropriate diversification strategy.

The use of exact methods would be the best choice, but the runtime heavily depends on the given problem instance. So we consider adapting an appropriate TSP heuristic method for the TSPTW (cf. Applegate, Bixby, Chvátal and Cook 2006).

REFERENCES

Applegate, D. E., Bixby, R. E., Chvátal, V., Cook, W. J., 2006. *The traveling salesman problem – a computational study*. Princeton:Princeton Series in Applied Mathematics.

Bräysy, O., Hasle, G., Dullaert, W., 2004. A multi-start local search algorithm for the vehicle routing problem with time windows. *European Journal of Operational Research* 159:586-605.

Clarke, G., Wright, J., 1964. Scheduling of vehicles from a central depot to a number of delivery points. *Operations Research* 12:568-581.

Cordeau, J.-F., Desaulniers, G., Desrosiers, J., Solomon, M. M., Soumis, F., 2001. VRP with time windows. In: Toth, P., Vigo, D., eds. *The vehicle routing problem*. Philadelphia:SIAM Monographs on Discrete Mathematics and Applications, 157-186.

Dantzig, G. B., Ramser, J. H., 1959. The truck dispatching problem. *Management Science* 6:80-91.

Dondo, R., Cerdá, J., 2007. A cluster-based optimization approach for the multi-depot heterogeneous fleet vehicle routing problem with time windows. *European Journal of Operational Research* 176:1478-1507.

Dorronsoro, B., 2007. *VRP Web*. Available from: http://neo.lcc.uma.es/radi-aeb/WebVRP/ [Accessed June 2008].

Gendreau, M., Potvin, J.-Y., 2005. Tabu search. In: Burke, E. K., Kendall, G., eds. *Search methodologies – introductory tutorials in optimization and decision support techniques*. New York:Springer Science+Business Media, 165-186.

Homberger, J., Gehring, H., 2005. A two-phase metaheuristic for the vehicle routing problem with time windows. *European Journal of Operational Research* 162:220-238.

Kallehauge, B., Larsen, J., Madsen, O., 2001. *Lagrangean duality applied on vehicle routing with time windows – experimental results*. Technical Report, IMM, Technical University of Denmark.

Laporte, G., Semet, F., 2001. Classical heuristics for the capacitated VRP. In: Toth, P., Vigo, D., eds. *The vehicle routing problem*. Philadelphia:SIAM Monographs on Discrete Mathematics and Applications, 109-126.

Potvin, J. Y., Kervahut, T., Garcia, B. L., Rousseau, J. M., 1996: The vehicle routing problem with time windows – part I: tabu search. *INFORMS Journal on Computing* 8:158-164.

Solomon, M. M., 1987. Algorithms for the vehicle routing and scheduling problems with time window constraints. *Operations Research* 35:254-273.

Thangiah, S. R., 1995: Vehicle routing with time windows using genetic algorithms. In: Chambers, L., eds. *Application handbook of genetic algorithms*. Florida:CRC Press, 253-277.

Toth, P., Vigo, D., 2001. An overview of vehicle routing problems. In: Toth, V., Vigo, D., eds. *The vehicle routing problem*. Philadelphia:SIAM Monographs on Discrete Mathematics and Applications, 1-23.

Wren, A., 1998. Heuristics ancient and modern: transport scheduling through the ages. *Journal of Heuristics* 4:87-100.

SENSOR NETWORK BASED CONFLICT RESOLUTION IN AUTONOMOUS MULTIAGENT SYSTEMS

Witold Jacak, [(a)] **Karin Pröll** [(b)]

[(a)] Department of Software Engineering
Upper Austria University of Applied Sciences
Hagenberg, Softwarepark 11, Austria
[(b)] Department of Bioinformatics
Upper Austria University of Applied Sciences
Hagenberg, Softwarepark 11, Austria

[(a)] Witold.Jacak@fh-hagenberg.at,[(b)] Karin.Proell@fh-hagenberg.at

ABSTRACT

The design of intelligent and sensor-based autonomous agents learning by themselves to perform complex real-world tasks is a still-open challenge for artificial and computational intelligence. In this paper a concept of a framework for an autonomous robotic agent is presented. The structure of an intelligent robotic agent consists of two independent subsystems: the action and motion planning system and the action and motion reactive control system with integrated conflict resolution methods. The action planning system uses an aggregated world model storing knowledge about all static and dynamic objects in the surrounding environment. The action controller solves space conflicts in a reactive manner making use of information from local sensors and a distributed sensor network. Each dynamic object registered from sensor network field is inserted into the world model as a new obstacle in 2,5D form. Based on the updated world model a conflictfree robot motion is calculated in a one step motion planning cycle.

Keywords: autonomous robotic agent, sensor-based conflict resolution

1. INTRODUCTION

The design of intelligent and sensor-based autonomous systems (agent type) that learn by themselves to perform complex real-world tasks is a still-open challenge for the fields of system and control theory, robotics and artificial and computational intelligence.

In this paper we present the concept of a framework for an autonomous robotic agent that is capable of showing both local sensor-based reactive behavior and global action planning based on external sensor network. Past experience has shown that neither purely reactive nor purely machine learning-based approaches suffice to meet the requirements imposed by real-world environments.

In multi-agent robotic systems, one is primarily interested in the behavior and interactions of a group of agents and a dynamic surrounded world, based on the models of the agents themselves and environment stimuli. With every perceptual input, one associates a certain action that is expressed in the form of rules or procedures that calculate the reaction of the agent. Reactive systems have no internal history or long term plans, but calculate or choose their next action based solely upon the current perceptual situation.

On the other hand, machine learning-based models are motivated by the representation of the system's knowledge. The adaptation of symbolic AI techniques has led to the introduction of believes and intentions into the reasoning processes of the system. Such models permit to use more powerful and more general methods than reactive models; this, however, makes them inadequate for many real-time applications where a dynamic change in the environment occurs.

Usually an agent has only partial information about the world state obtained from own perception system (sensors system). Machine learning aims to overcome the limitations such as knowledge bottleneck, engineering and tractability bottleneck, by enabling an agent to collect its knowledge on-the-fly, through real-world experimentation. Processing, storing and using information obtained during several task executions is called lifelong learning. For this reason it is necessary to extend the reactive control system of sensor-based global preplanning system by omitting the knowledge bottleneck in classical machine learning approach.

2. CONFLICT RESOLUTION SYSTEM FOR ROBOTIC AGENT

In our concept, the structure of an intelligent robotic agent consists of two independent subsystems: the action and motion planning system and the action and motion reactive control system with integrated conflict resolution method (Jacak, Proell, and Dreiseitl 2001, Jacak and Proell 2007). The action planning system uses an aggregated world model storing knowledge about all conflicts which occurred in the past. Conflicts occur when a collision between a robotic agent and an unknown dynamical object or another agent in work space is possible. The action controller is able to solve the conflict situation (space conflict for autonomous robotic agent) in a reactive manner. The general conflict resolution part of the agent makes use of knowledge not only from sensors mounted on active agent (local sensor network) but also from distributed global sensor network (see Figure 1). A sensor network is a collection of sensor nodes deployed in an adhoc fashion in the work space.

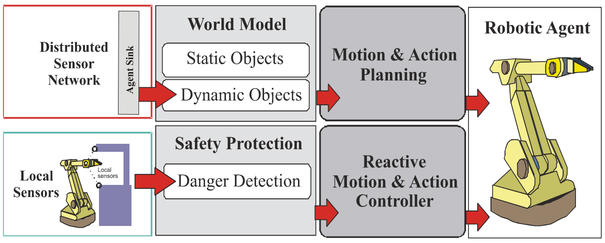


Figure 1: Structure of sensor-based conflict resolution system

Being battery powered and deployed in remote areas they have limited energy resource and hence limited lifetime. Other constraints include limited memory, processing power, and band-width. The accuracy of information is location dependent. Due to these limitations data aggregation is an important consideration for sensor networks. The idea is to combine the data coming from different sources and reroute it further, after eliminating redundancy, minimizing number of transmissions and thus saving energy.

Sensors will be used for inventory maintenance and unknown object recognition and motion tracking. The agent requests a monitoring of some segment of space in which the next parts of motion are supposed to take place from the global sensor network. If the global sensor network identifies unknown objects in these segments then this information can be used by the agent’s safety system to preplan a reaction prior to recognition of these objects by the local perception units. This can help to avoid possible conflict situations in advance.

3. SENSOR NETWORK BASED ONE STEP MOTION PLANNING SYSTEM

3.1. Static and Dynamic World Model

The knowledge represented here is the geometrical model of the robotic agent environment. Many different methods can be used for geometrical representation of the agent service space. One of them is the triangle approximation another is cubic approximation.

In a model with triangle approximation, the points in the triangle net (lying on service space border) are coupled in triangle walls. The walls represent the data objects of the world model.

The other model describes the service space of the robot manipulator as cubic approximation. The service space of robotic agent can be discretized in the form of the cubic raster (voxels). The number of voxels depends of the accuracy of approximation. If the voxel is occupied by an obstacle then it obtains the value 1 of the space occupancy function.

The level of fullness of knowledge leads to two different methods of resolving of conflict situation. The both models are convenient to represent geometrical environment of known objects in agent workspace. The model is used for collision-freeness testing between agents and surrounding world. We can say that the position of an agent is collision free if it does not collide with any static or dynamic obstacle in its workspace. To test such conditions we should have full knowledge about the surrounding static and dynamic world, that means that the geometrical model of agent’s environment should be completely known. To obtain fast and fully computerized methods for collision detection, we use additional geometric representation of each object on the scene.

We introduce the ellipsoidal representation of 3D objects, which uses ellipsoids for filling the volume. The ellipsoidal representation of an object is convenient to test collision freeness of agent positions. The 3D models of objects represent the static part of agent’s world model. To construct the model of unknown objects, which penetrate the agent’s environment, it is necessary to continuously modify the geometrical representation. The dynamic part of model is modified based on sensor data coming from wireless sensors network.

3.2. Sensor Network

A sensor network is composed of a large number of tiny autonomous devices, called sensor nodes. Each sensor node has four basic components: a sensing unit, a processing unit, a radio unit, and a power unit. Since a sensor node has limited sensing and computational capabilities and can communicate only within short distances. The nodes are deployed densely and coordinate amongst themselves to achieve common information (Karl and Willig 2005). Some examples of sensor network applications are as follows:

- *Intrusion detection and tracking*: Sensors are deployed along the border to detect, classify, and track intruding objects (Dousse, Tavoularis, and Thiran, 2006).
- *Environmental monitoring:* Specialized sensor nodes that are able to detect changes in environment (Tzung-Shi Chen, Yi-Shiang Chang, Hua-Wen Tsai, Chih-Ping Chu 2007).

These sensor networks applications differ significantly. However, the tasks performed by the sensors are similar: sensing the environment, processing the information, and sending information to the base station(s). A node in a sensor network has essentially three different tasks (Al-Karaki and Kamal 2004):

1. Sensing: detecting changes of environment;
2. Communicating: forwarding information, acting as an intermediate relay in a path;
3. Computing: data aggregation, processing, and compression.

Routing techniques are needed to send data between sensor nodes and the base station. Several routing protocols are proposed for sensor networks. These protocols can be divided into the following categories: data-centric protocols, hierarchical protocols, location based protocols, and some QoS-aware protocols (Dousse, Tavoularis, and Thiran, 2006).

For monitoring the surrounding environment of the robotic agent we propose a sensor network based on a virtual grid representation of the monitored area and for data delivery a combination of event driven and query-driven data delivery protocol between sensor network and a mobile sink component (AS) of the robotic agent is used (Tzung-Shi Chen, Yi-Shiang Chang, Hua-Wen Tsai, and Chih-Ping Chu 2007). The mobility of AS results from movements of the robotic agent.

Virtual Grid Structure of Sensor Network: The monitored area is divided into virtual grids. We notate *G(x,y)* as the grid coordinates, where *x* is grid x-coordinate and *y* is grid y-coordinate in Cartesian space. Let *R* be the transmission distance of the radio signal and *d* be the side length of grid. After sensors have been deployed, a node is selected to act as grid head. The grid head's task is to record information about events and disseminate it to other nodes for collaborative signal and information processing. At first, each node obtains neighboring information by start message. Utilizing this information, a node that is closest to the center of the grid is selected as head. If a head has not enough resources, one of other nodes in the same grid will be selected to replace it.

We assume that the side length of grids to guarantee that the grid heads can communicate with neighboring grids directly. Here, we also assume that the communication range of node is able to communicate with the neighboring grids. The relationship between *d* and *R* is predefined.

The grids are grouped in sub-networks in hierarchical way. Each sub-network obtains one head selected from the set of grid's heads. The sub-network head collects the messages from grid's heads within the sub-network.

The mobile agent sink (AS) stores the topology of the virtual grid and uses this information to perform the queries to the sensor network.

The task of AS is to find the virtual cell of grid where the intrusion of a new dynamic object is registered. An AS expects to obtain the event information instantly when such event occurred. An event usually happens unexpected. Therefore, a sensor has to signal an event after it was detected. This sensor is called source node. The source node propagates a register packet to all grid heads. The format of register packet is <*P_type, Src_id, Src_G(x,y), hc, event_type, time_to_expire*>, where *P_type* is packet type, *Src_id* is the identifier of the source node, *Src_G(x,y)* is the source's grid location, and *hc* is the hop count. When heads receive this packet, they store the register packet in their register table and route it to the sub-network head. This information is kept within a certain period of time (*time_to_expire*). If a head does not receive any further register packets and *time_to_expire* is elapsed, it removes the information from the register table.

The information of an event is distributed to the grid heads and summarized to the sub-network head in the following way: The grid heads store the *x,y* positions of active sensor nodes (an event has been signaled) whereas in the subnet-work heads only grid numbers with active sensor nodes are registered (active grids).

For the next query the AS maintains a list containing all sub-network heads surrounding the next segment of its motion path and all sub-networks with active grids.

AS can now decide to obtain detailed information about all active sensor nodes by querying the grid heads of all active grids or summarized information about all active grids by querying only the sub-network heads. For reconstructing the geometric model of the intruding object both - detailed or summarized - information can be used. Using detailed information a more accurate shape of the base of the intruding object can be deduced. The use of summarized information leads to a very approximate representation based on shapes of active grids. (see Figure 2)

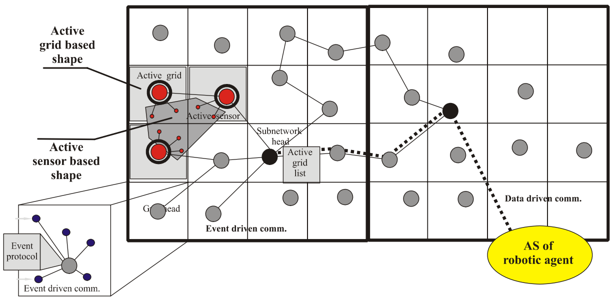


Figure 2: Use of summarized or detailed information from virtual grid for reconstructing shape of intruding object

By maintaining a dynamic list of sub-network heads for querying, a general broadcast to all sub-network heads in the virtual grid can be avoided. The possibility to query either sub-network heads or grid heads further reduces transmission activities in the network.

World Model Updating: The path planner of robotic agent sends a query to AS to check if any grid cell in sensor network has an active signal of a new object intrusion. AS uses the sub-network heads and grid heads and hierarchical protocol to collect the $G(x,y)$ position of grids to be active. The $G(x,y)$ and side length d will be used to approximation of object shape and volume to be intruded in the work space of robotic agent. The 3D geometrical model of an intruding object is constructed in a 2,5D representation of the shape registered by sensors (see Figure 3).

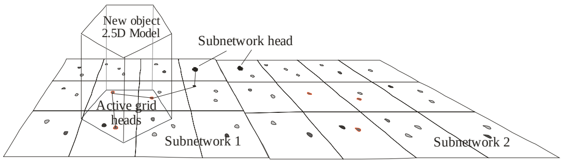


Figure 3: Virtual Grid Structure of Sensor Network and 2,5 D representation of intruding object

3.3. Global Motion Planner

Agent Model: The most suitable model of hardware component of robotic agent is a discrete dynamic system. One of the ways to construct such model of robot's kinematics is based on an arbitrary discretization of angle increments of the agent mechanical joints (Jacak 1999). Using the fact that all the angles can change only by a defined increment, we define the input set U of the model as: $U = \times\{u_i | i=1,..,n\}$ where $u_i = \{-\delta q, 0, \delta q\}$ is the set of possible (admissible) directions of changes of the *i-th* joint angle. Having defined the set *U*, it is possible to describe the changes of successive configurations of the agent's link as discrete linear system with the state transition function as:

$$q(k+1)=q(k) + \Lambda\, u(k) \qquad (1)$$

In order to make it possible to check the configuration with respect to obstacles locations, it is necessary to create an output function q. A skeleton of agent´s arm represents the agent position in the base frame. The i-th joint's position in Cartesian base space (skeleton point), assuming that all the joint variables q are known, is described by the Denavit-Hartenberg matrix. Checking for the collision-freeness of the agent configuration (skeleton) can be reduced to the "brokenline ellipsoid" (skeleton ellipsoid) intersection detection problem, which has an easy analytical solution. The complete formal explanation of the FSM model of agent kinematics, is presented in (Jacak 1999, Proell 2002).

Motion Planner: For the robotic agent we can define the problem of achievement the goal position as the problem of reachability of the final state set from the agent's current state (current position).

In order to solve this problem we apply graph searching in the state transition graph. The process of applying the transition function to a current state we term expanding the graph node. Expanding current state *qc*, successors of *qc* etc. ad infinitum, makes explicit the graph that implicitly is defined by current state and transition function. The way of expanding the graph will depend on the form of the cost function using to evaluate each node. As the evaluation function we can use the sum of the cost function $c(qc,q)$ and a cost estimate function $h(q,qf)$, for example the rectilinear distance between agent position and terminal position *qf*.

Using the standard A* procedure we can find the state trajectory (if exists) $q^*=(qc, q(2),..,q(k),.., qf)$ from current state to final state *qf* which includes only feasible states. In order to solve the path-planning problem we apply the graph searching procedure to the agent's state transition graph. The development of the search graph will start from the node (configuration) *qc*, by the action for all possible inputs from the set U. Thus, it becomes essential to quickly check for the non-repeatability of the nodes generated, and their feasibility. A configuration *q* is said to be feasible if it does not collide with any object in the surrounding world.

3.4. One step motion path planning

The continuously adapted world model is used to test the collision freeness. Each dynamic object registered from sensor network field is inserted into the world model as a new obstacle in 2,5D form.

The task of the motion planning and execution component is to plan the collision free configuration of the robot's manipulator based on information coming from the world model from knowledge base. To realize this task, the on line motion planner calculates the changes of robot configuration to avoid the obstacle in the best possible way. This knowledge is represented as geometrical model of work scene and mathematical model of agent's actor direct and inverse kinematics.

The one step action planner of the agent generates the new movement in following steps:

step 1: The motion planner requests the current world model from AS of sensors network,

step 2: Based on the positions of obstacles motion planner recognizes its own current position and establishes the parameters for motion search algorithm, such as type of evaluation function.

step 3: Action planner starts the stategraph searching algorithm A* from its current position with previously established evaluation. The searching stops if the goal position is reached or if the OPEN set is empty or if the OPEN set has more as N elements. To calculate the successors set planner uses the FSM - model of the agent.

step 4: The temporary path of motion is calculated and the new configuration of motion is chosen and realized. The motion planner started step 1 again.

3.5. Example

The following example (Figure 4) shows the trajectories of the robotic agent at times T_1, T_2, $\ldots T_8$ avoiding the cylindrical object crossing its initial path. At each point of time T_i the motion planner uses the information about the obstacle's current position from the sensor network for recalculating a new collision free motion as a temporary path from current to final position. When the object leaves, the robotic agent does not return to its initial trajectory at time T_1 but takes the shortest way from the current to the final position.

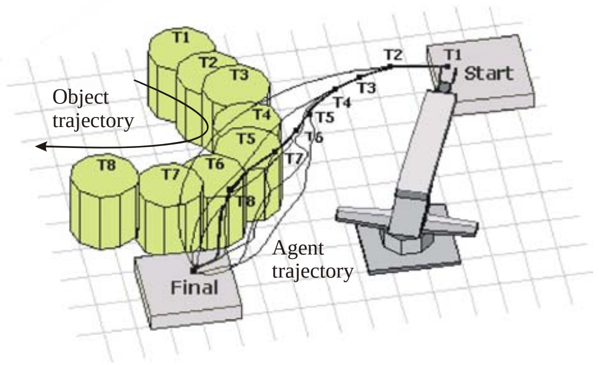


Figure 4: One step motion planning to avoid collision with cylindrical object

4. REACTIVE LOCAL CONTROLLER OF MOTION

4.1. Neural network based agent model

The agent model contains the knowledge about construction, properties and structure of a hardware agent. Here, knowledge of the kinematical properties of the robotic agent is provided in order to decide about collision avoidance mode and avoidance path. Therefore, the forward and the inverse kinematics models of the robot should be known.

The planning of the new configuration is based on the computation of robot kinematics and is computationally expensive. Therefore it is attractive to apply a neural network model of robot kinematics, stored in the knowledge base of the agent, which automatically generates safe configurations of the robot. This model uses a multilayer feedforward neural network with hidden units having sinusoidal activation functions (Jacak 1999).

Each element of manipulator direct kinematics can be represented in the form of $\Pi^{i=1}_{n}\ a_i\ sc(q_i)$ where $sc(q_i)$ is either $\sin q_i$, $\cos q_i$ or 1. Then, after simplification, the forward kinematics of a robot manipulator with n revolute joints can be described by the weighted sum of sinusoidal functions: $t_i^s(q)=\Sigma \sin(w_j^T q)$ and $s=x,y,z$ where $t_i^s(q)$ defines the output of neural network representing s-th Cartesian variable of i-th joint position. Based on the neural direct kinematics model it is easy to generate the inverse kinematics (Jacak 1999, Proell 2002). The plant model of the robot kinematics is presented in Figure 5.

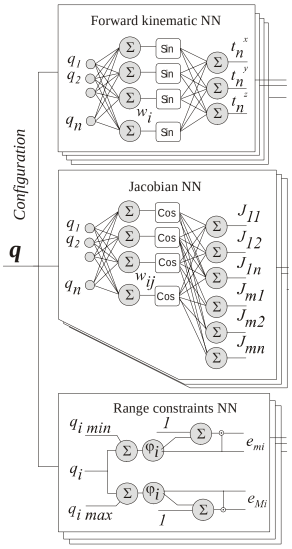


Figure 5: Neural network model of the robot kinematics

4.2. Safe local motion planning and execution

To calculate the safe configuration, the planner uses information from the local sensors of danger recognition component and combines it in the inverse kinematics computation. We use the method, which requires only the computation of direct kinematics based on neural processing. Such the solution of inverse kinematics problem can be obtained by attaching a feedback network around a forward to form a recurrent loop, such that, given a desired Cartesian pose P of a feedforward network, the feedback network iteratively generates joint angle correction terms to move the output of forward network toward the given pose. This coupled neural network is the neural implementation of a gradient method of position error minimization.

Let $e_n(q)$ denote the position-error between the current position of effector-end in current configuration q, calculated by forward kinematics and a desired next Cartesian position on the executed path. The obstacle avoidance can be achieved by introducing the additional errors for each joint, i.e. the errors between virtual points p_i and the joints positions, where the virtual points represent the wished position of the i-th link that achieves collision-avoidance. The virtual points are placed on the opposite side of the joint with respect to the obstacle. The virtual points are calculated based on signals from local sensor system, mounted directly on the robotic agent. To solve the inverse kinematics problem in our particular case we transform it to the optimisation problem. Note, that the solution of the inverse kinematics problem uses only direct kinematics models, and Jacobian. Both models can be implemented as neural networks (Jacak 1999).

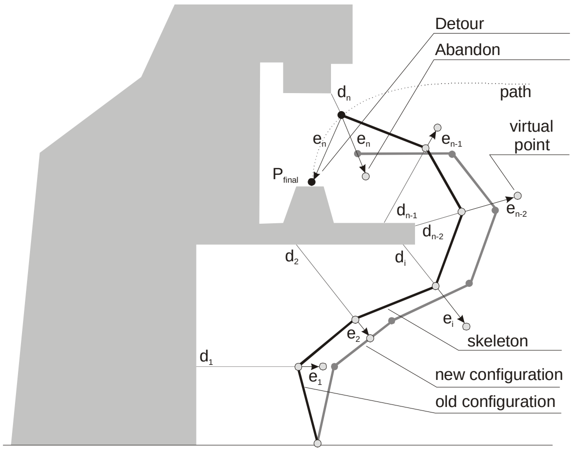


Figure 5: virtual points and reactive motion control

For calculation of virtual points, we propose the installation of ultrasonic sensors and a sensitive skin on each manipulator link and then use a neural network to estimate the proximity of the objects with the link of question. The resulting distances are compared with the world model of the robot environment to recognize the new obstacle within the radius of the security zone ρ.

When a manipulator is equipped with many sensors and these sensors are mounted on different manipulator links the problem arises how to fuse sensors readings to obtain useful information. The results of the fusion action are vectors $d=(d_i|i=1,..,n)$, representing the minimal distances to objects in robot environment for each joint of the manipulator. The distances are used to calculate the virtual points for online inverse kinematics method. When the obstacle penetrates the local security zone the virtual points are calculated and additional errors are introduced into inverse kinematics. The reactive motion controller calculates the new position of agent which minimizes additional errors i.e. maximizes the distances d to obstacles. (see Figure 5)

CONCLUSION

Using global sensor network unknown objects in a robotic agent's workspace can be identified prior to recognition of these objects by the local perception units. This information is stored in the world model of the robotic agent which is continuously updated. The aggregated world model is used by the agent's safety system to preplan a reaction to avoid possible conflict situations. Especially for mobile robots with a manipulator fixed on top of a mobile base unit the combination of a global sensor network and local perception units improves their mobility.

REFERENCES

Al-Karaki, J.N. Kamal, A.E. 2004. Routing techniques in wireless sensor networks: a survey. *Wireless Communications, IEEE* Volume 11, Issue 6:6- 28.

Dousse, O., Tavoularis, C., and Thiran, P. 2006. Delay of intrusion detection in wireless sensor networks. *Proceedings of the 7th ACM international Symposium on Mobile ad hoc Networking and Computing*, 155-165. May 22 - 25, 2006, Florence, Italy.

Jacak W., Proell K., 2007. Heuristic Approach to Conflict Problem Solving in an Intelligent Multiagent System Heuristic Approach to Conflict Problem Solving in an Intelligent Multiagent System. In: *Computer Aided Systems Theory — EUROCAST 2007, Lecture Notes in Computer Science*. Heidelberg:Springer, 772-779.

Jacak W., Proell K., Dreiseitl S., 2001. Conflict Management in Intelligent Robotic System based on FSM Approach. In: *Computer Aided Systems Theory — EUROCAST 2001, Lecture Notes in Computer Science.* Heidelberg:Springer, 359-386.

Jacak, W., 1999. *Intelligent Robotic Systems: Design, Planning and Control.* New York, Boston, USA: Kluwer Academic/Plenum Publishers.

Karl H, Willig A., 2005. *Protocols and Architectures for Wireless Sensor Networks*. USA: Wiley.

Proell K., 2002. *Intelligent Multi-Agent Robotic Systems: Contract and Conflict Management*. PhD Thesis, Johannes Kepler University Linz /Austria.

Tzung-Shi Chen, Yi-Shiang Chang, Hua-Wen Tsai, Chih-Ping Chu, 2007. Data Aggregation for Range Query in Wireless Sensor Networks. *Journal of Information Science and Engineering* Volume 23, Number 4:1103-1121.

MODELLING OF IMMUNE FUNCTIONS IN A WIRELESS SENSORS NETWORK

Jan Nikodem [a], Ryszard Klempous[a], Zenon Chaczko[b]

[a]Wroclaw University of Technology, Poland
[b]University of Technology Sydney, NSW, Australia

[a] Jan.Nikodem@pwr.wroc.pl, [b] Ryszard.Klempous@pwr.wroc.pl, [c] Zenon.Chaczko@uts.edu.au

ABSTRACT

In this paper we explore relationships between immunity and adaptation in Wireless Sensors Network (WSN). We consider the WSN as a set of sensors deployed in a given area. The sensors must communicate to achieve both their particular self-interests and global goals. In the proposed approach we determine immunity and adaptation abstractions and considering them in the context of three fundamental relations (subordination, tolerance, collision). The proposed conceptual framework provides a powerful paradigm to conceptualise, model, support and manage dynamically organizing complex systems processes.

Keywords: artificial immune system, immunity of WSN, relations in complex system

1. INTRODUCTION

The research presented in this paper aims to focus on the Wireless Sensors Network (WSN) that consists of homogeneous sensors cooperating with each other. WSN is not organised centrally, but rather in a distributed manner.

Each individual sensor has limited resources such as: energy, hardware, software and communication range, hence in the individual level it is not able to realise overall system tasks. Due to these limitation, sensors are required to lesser or greater degree strictly follow an integrated cooperation between them. This activity is realised predominately in the information domain, as it is essential for the WSN dependability to provide a robust communication capability.

From one point of view, each sensor works in its precinct (vicinity) autonomously, interacting with environment stimulus; And from the other point of view, sensors must communicate with each other, therefore communication channels are crucial elements of the WSN architecture. In general, sensors are simple, unsophisticated technical components performing tasks determined by programmers and engineers. Because of that the risk of unauthorized access or even a possible destruction of communication links by outsiders can be an increasing threat. External attackers are often responsible for causing communication and routing disruptions including the breach of security. Frequently, however, it is the internal events which may contribute to serious decrease in efficiency of communication channels.

2. BASIC IDEAS AND RATIONALE

When describing the WSN activity we will be discussing the concepts of actions and behaviour. Action should be considered the property of every network element such as: a sensor, a cluster head or a node. The Behaviour, on the other hand is an external attribute which can be considered either as an outcome of actions performed by the whole WSN or its subset (i.e. cluster, tree, sensor field, vicinity). *Action* is a ternary relation which can be defined as follows:

$$Act : Nodes \times State \rightarrow State. \tag{1}$$

Based on this we can construct the quotient set *Behaviour*, elements of which are called equivalence classes linked to the relation *R* and here denoted as:

$$Beh : Act / R = \{act \in Act \mid act\ R\ x\}. \tag{2}$$

2.1. Neighbourhood abstraction

Let us define *Map(X, Y)* as a set of mapping functions from *X* onto *Y* (surjection). Where *Sub(X)* is defined as a family of all *X* subsets. We define the neighbourhood $\mathcal{N}$ as follows:

$$\mathcal{N} \in Map(Nodes, Sub(Nodes)). \tag{3}$$

Thus, $\mathcal{N}(k)$ is the neighbourhood of node *k*, and $\mathcal{N}(C)$ is the neighbourhood of *C* (set of nodes) defined below as:

$$\mathcal{N}(k)\big|_{k \in Nodes} := \{ y \in Nodes \mid y R_{\mathcal{N}} k \}, \tag{4}$$

$$\mathcal{N}(C)\big|_{C \subset Nodes} := \{ y \in Nodes \mid (\exists x \in C)(y R_{\mathcal{N}} x) \}. \tag{5}$$

The formal view emerging from the above discussion will project a construction of the extended WSN behavioural model and related to the challenges of individual tasks versus collective activities of network elements. By *Individual* it is meant that the node/sensor improves its goal-reaching activity, interacting with its neighbourhood $\mathcal{N}(k)$ while collective activity relates to $\mathcal{N}(C)$ neighbourhood.

2.2. Relational attempt to network activity

In 1978 J. Jaron has developed an original methodology for the systemic cybernetics (Jaron 1978) which describes three basic relations between systems components such as: subordination (π), tolerance (ϑ) and collision (χ). We find these basic relations very useful in describing activities and qualitative relations between components of the WSN. For concepts presented in this paper, and for discussion on the immune functions in communication activities, we require to refer the following three key relations:

$$\pi := \{< x, y >; x, y \in Act \mid x \pi y\}. \tag{6}$$

The expression $x\ \pi\ y$ - defines the action x which is subordinated to the action y or action y dominate over action x.

$$\vartheta := \{< x, y >; x, y \in Act \mid x \vartheta y\}. \tag{7}$$

The expression $x \vartheta y$ - states that the actions x and y tolerate each other,

$$\chi := \{< x, y >; x, y \in Act \mid x \chi y\}. \tag{8}$$

And finally $x\chi\ y$ - means the actions x and y are in collision to one another. The basic properties of mentioned above relations could be formulated succinctly as follows (Jaron, 1978):

$$\pi \cup \vartheta \cup \chi \subset Act \times Act \neq \emptyset, \tag{9}$$

$$\iota \cup (\pi \cdot \pi) \subset \pi, \tag{10}$$

where $\iota \subset Act \times Act$ is the identity on the set *Action*.
Moreover,

$$\pi \cup \vartheta^{-1} \cup (\vartheta \cdot \pi) \subset \vartheta, \tag{11}$$

where ϑ^{-1} is the converse of ϑ so,

$$\vartheta^{-1} := \{< x, y > \in X \times Y \mid y \vartheta x\}. \tag{12}$$

For collision,

$$\chi^{-1} \cup (\pi \cdot \chi) \subset \chi \subset \vartheta'. \tag{13}$$

where ϑ' is the complement of ϑ so,

$$\vartheta' := \{< x, y > \in X \times Y \mid < x, y > \notin \vartheta\}. \tag{14}$$

The formula (9) indicates that all these three relations are binary on nonempty set *Actions*. The formula (10) describes fundamental properties of subordination relation which is reflexive and transitive. Therefore it is also ordering relation on the set *Actions*.

The formula (11) states that subordination implies the tolerance. Hence we can obtain:

$$(\forall x, y \in Act)(x \pi y \Rightarrow x \vartheta y) \tag{15}$$

and subordinated actions must tolerate all actions tolerated by dominants

$$(\forall x, y, z \in Act)((x \pi y \wedge y \vartheta z) \Rightarrow x \vartheta z). \tag{16}$$

3. MODELLING OF IMMUNE FUNCTION

Immunity is the distributed rather than the global property of a complex system. The complexity of the immune function, the absence of simple feedback loops, a high complexity of tasks and activities; and collective (not central) rather than individual regulatory processes all results in serious challenges for attempts to define system's immune competences.

There are a number of fundamental works corresponding to this domain and related to artificial systems (Hofmeyr, Forrest, 2000), (Timmis, 2000). Some authors have drawn inspiration from the biological immune system, incorporated a lot of properties from autonomous immune systems.

In this article, we propose a novel relational approach to modelling of the system immunity. In our attempt we are considering the immunity not as a set of particular mechanisms like clonal selection, affinity maturation or elements (antigens, antibodies, idiotopes, paratopes, etc.). In the proposed approach the phenomenon of immunity appears to be a result of inter-relations among members of community (neighbourhood).

Let us consider for the node k, its neighbourhood $\mathcal{N}(k)$. Any communication activity act_k that is performed by node k relates to some members of $\mathcal{N}(k)$ and the set of actions act_k within neighbourhood $\mathcal{N}(k)$ can be defined as follows:

$$Act_{\mathcal{N}}(k) := \{act_k \in Act \mid (\exists x \in \mathcal{N}(k))(act_x R\, act_k)\}. \tag{17}$$

If n is a number of actions act_k within neighbourhood $\mathcal{N}(k)$, then it can be expressed as cardinality *Card* ($Act_{\mathcal{N}}(k)$). The collection of all subsets of $Act_{\mathcal{N}}(k)$ is determined as power set $Pow(Act_{\mathcal{N}}(k))$ with cardinality 2^n. Finally, any subset of that power set is called a family *Fam* ($Act_{\mathcal{N}}(k)$) of communication activity within neighbourhood $\mathcal{N}(k)$.

The Cartesian product defined as:

$$IS_k := Act_{\mathcal{N}}(k) \times Act_{\mathcal{N}}(k) \subseteq \pi \cup \vartheta \cup \chi \tag{18}$$

describes interaction space IS_k within $\mathcal{N}(k)$.

Let us now consider z_k=Card(IS_k) which defines a number of possible interactions within neighbourhood N(k) which can be expressed as:

$$R_k := \{ y \in Act_{\mathcal{N}}(k) \,|\, < k, y > \in IS_k \wedge kRy \}. \qquad (19)$$

Thus, for a given relation *R* we define intensity quotient within neighbourhood $\mathcal{N}$(k) as follows:

$$IR_k = Card(R_k)/z_k. \qquad (20)$$

The rationale behind our choice of the relational approach to modelling immune functions is the fact that interactions are factual and a very relevant aspect of elements of the immune system. The immune system itself can be defined as a very large complex network with a layered and hierarchical architecture. Relationships between components of this architecture can be described by the subordination relations (π). Furthermore, a positive response of the immune functions would result from the collision relation (χ) while a negative response would be attributed to tolerance (ϑ) relation.

An immune system is adaptive in nature in the sense that its positive or negative responses should be adequate to environment stimulus. Since resources are limited, the allocation rule predicts that an increased investment in system adaptability will come at a cost to investment in immunity (and vice versa). Growing adaptability decreasing system immunity and extends possibility of adversary's attacks. On the other hand, growing immunity barrier tends to adaptability reduction. Bellow we define immunity and adaptability as follows;

$$immun := \{ < x, y >; y, x \in Act \,|\, < x, y > \in \vartheta' \cup \chi \}, \qquad (21)$$

$$adapt := \{ < x, y >; y, x \in Act \,|\, < x, y > \in \vartheta \cup \pi \}. \qquad (22)$$

Therefore, the scope of immunity is determined by decreasing tolerance and growing collision relations when the extend of adaptation is determined by expanding tolerance and subordination.

Additionally, based on (21), (22) it is possible to model adaptation - immunity characteristics of a system and consider it as a process of finding a fine homeostatic balance between them. In our case the set of feasible solutions is modelled as a line segment (Figure 1) linking strong immunity *(SI)* and strong adaptability *(SA)*. This line is obtained as the intersection of a plane [*SA*, (1,1,0), *SI*, (0,0,1)] with a plane [(0,0,0), *SI*, (1,1,1), *SA*)]. In such approach it is practically impossible to determine balance point *B*∈ *(SI; SA)* which corresponds to both global and any local situations. Based on relational approach (21), (22), it is evident (as shown in Figure 1) that the issues of keeping the balance between adaptation and immunity is far more refined then simply finding a point on line segment *(SI; SA)*. This line segment is a canonical projection of topological space subset (tetrahedral on Figure1). Any point at this solid figure represents one of many, feasible solution which can fulfil (for better or worse) the quality requirements of the network system.

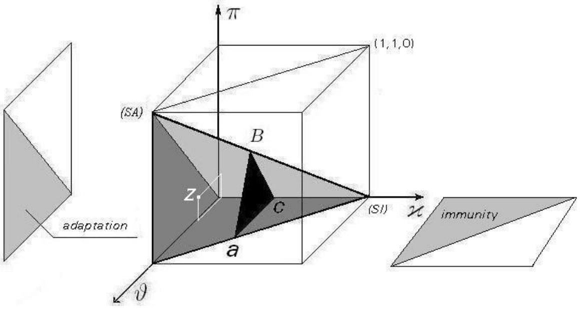
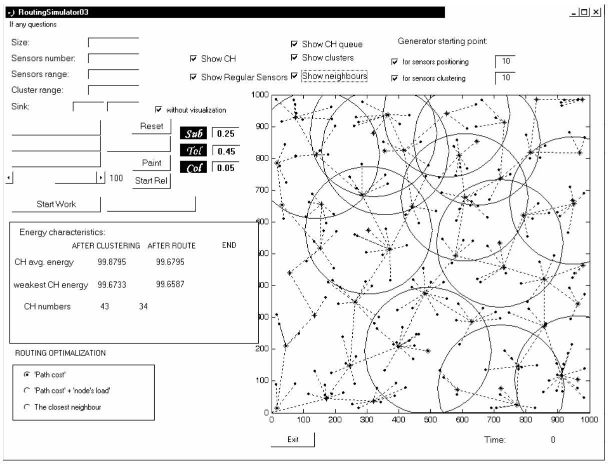


Figure 1: Modelling immunity based on π, ϑ and χ relations.

Choosing the point *B*∈ *(SI; SA)* we determine globally the desirable adaptability/immunity homeostatics. However, for each node, owing to its neighbourhood state, it is necessary to determine an individual point of balance. The set of such points has to be determined with reference to equivalent class (triangle *aBc* in Figure 1) that corresponds exactly to one point *B* (Figure 1) on a line segment *(SI; SA)*.

4. HOW IT WORKS IN PRACTICE

In order to illustrate the modelling immunity result with a concrete model simulation, consider WSN with 10 nodes and one base station (BS). To determine neighbourhood $\mathcal{N}(k)$ assignment we use radio link range. Therefore, coming back to the (3)-(4) we create subsets $\mathcal{N}(k)$, *k=1, 2.., 10.* Each cell *n* of row *k* in the binary matrix $\mathcal{N}$ (Figure 4) represents a membership *n*∈ $\mathcal{N}(k)$. It is worth mentioning that a neighbourhood always related to communication range, but very rare corresponds with a structure of clusters. Such constructed neighbourhood subsets are unambiguous and stable as long as a communication range is fixed.



☑ Show CH
☑ Show Regular Sensors
☑ Show CH queue
☑ Show clusters
☑ Show neighbours
☑ without visualization

Generator starting point:
☑ for sensors positioning 10
☑ for sensors clustering 10

Sub 0.25
Tol 0.45
Col 0.05

Energy characteristics:

	AFTER CLUSTERING	AFTER ROUTE	END
CH avg. energy	99.8795	99.6795	
weakest CH energy	99.6733	99.6587	
CH numbers	43	34	

ROUTING OPTIMALIZATION
● 'Path cost'
○ 'Path cost' + 'node's load'
○ The closest neighbour

Figure 2: WNS activity simulator based on relational space (π, ϑ and χ relations).

In the context of WSN activity, we focus our attention on routing aspects. Therefore in furthers consideration, from all action sets $Act_{\mathcal{N}}(k)$ within neighbourhood k we select only these, performed routing activity. In such way we obtain a family *Fam* $(Act_{\mathcal{N}}(k))$ of routing activity. Considering the distance from the base station (BS) to a node position, a routing activity within $\mathcal{N}(k)$ is partially ordered (≤). Additional preferences were given to cluster heads (CH) and those nodes which belong to routing tree. Firstly, a cluster heads on routing path are the nearest BS, next another BS, finally the other nodes within $\mathcal{N}(k)$ (Figure 3).

In order for us to proceed any further, it is necessary to combine the WSN spatial structure (topology) and the WSN activity. To facilitate this process we are required to construct a product that describes the interaction space IS_k within $\mathcal{N}(k)$ as defined in (18). Each element of the interaction space (regardless of which neighbourhood we consider), according to the right side of (18) equation is mapping (as a point) to (may be not injection) the 3D relational space [π, ϑ, χ] as presented in Figure.1 above.

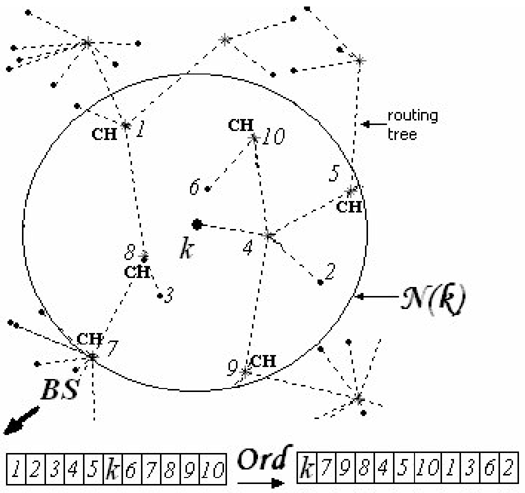


Figure 3: Modelling partially ordered neighbourhood abstraction $\mathcal{N}(k)$.

In order to model [π, ϑ, χ] space we exploit three additional matrices presented on Figure 4. The real number in any cell of these matrices expresses an intensity quotient of relation. Elements [$r_{k,k}$] represent intensity quotient within $\mathcal{N}(k)$, while [$r_{i,j}$] related to particular *i-j* nodes interactions. Henceforth, we are ready to model the dichotomies of the WSN characteristics - immunity and adaptability.

The initial step is the determination of the global (for whole network) strategy GS_{WSN} for the network activity. For this aim, we construct a subset of relational space (9) as a conjunction of (10), (11), (13) with (21), (22). This subset (depicted as tetrahedral in Figure 1) consists of all feasible actions. The derived set constitutes a global interaction space IS_{WSN} (18) within the WSN network. Now we shall determine global strategy GS_{WSN} as a subset of interaction space IS_{WSN}. Notice that there are a huge number of different choices of such subset, but for simplicity reason we consider only two. First is the subset:

$$GS^1_{WSN} = \{ y \in IS_{WSN} \mid y \in \Delta aBc \}, \tag{23}$$

where, ΔaBc - is a black triangle on Figure1, second is a white point Z on the same Figure 1. While first is a restriction from 3D to 2D, the second is restriction from 3D to 1D mapping. Clearly, any restriction of IS_{WSN} space offers less choices then the whole, but as we have shown bellow even in case of singleton

$$\{Z\} = \{\langle \pi_k, \vartheta_k, \chi_k \rangle\} = \{\langle 0.2,\ 0.54,\ 0.07 \rangle\} = GS^2_{WSN} \tag{24}$$

the spectrum of possible activities is rather wide.

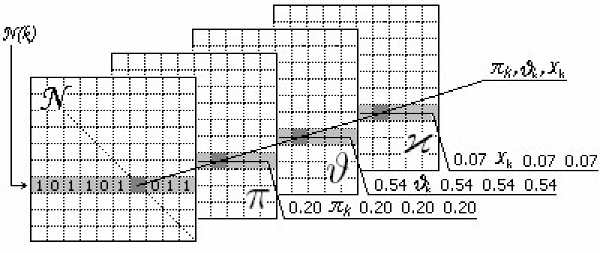
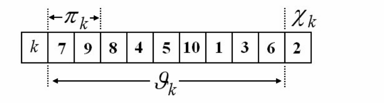


Figure 4: Modelling relational *(π, ϑ, χ)* space within $\mathcal{N}(k)$ neighbourhoods.

In the following step we shall determine local strategy for each neighbourhood $\mathcal{N}(k)$, $k=1, 2.., 10$. In order to deal with the case where the global strategy is determined by (24) the local strategy for neighbourhood activity need to be identical. Hence for each node k its interaction space IS_k is a singleton $\{Z\}=\{<\pi_k, \vartheta_k, \chi_k>\}$ exactly the same like (24) and the real numbers π_k, ϑ_k, χ_k occupied main diagonal cells of relational matrices π, ϑ, χ (Figure 4). Henceforth, the proper local (within neighbourhood $\mathcal{N}$(k)), activity can be accomplished in accordance with the global strategy. Thereby, instead to force the local activity of each node, we are rather formulating requirements for the nature and intensity of relations within $\mathcal{N}$(k)). Fulfilment of these requirements results in desired global strategy.

In the last step of preparing the simulation process we shall determine non-diagonal elements of relational matrices (Figure 4). To facilitate the representation of nodes interactions (point-to-point) we assign them a real numbers exactly the same like on diagonal. The starting point of simulation determined in such way is feasible of course (holds both local and global strategies).

The relational state for any $\mathcal{N}$(k) is now identified with ordered sequence of node indexes as follows:



$$\langle \pi_k, \vartheta_k, \chi_k \rangle = \langle 0.2, 0.54, 0.07 \rangle. \qquad (25)$$

Considering that a given WSN consists of 10 nodes, we say there is ten different ordered sequences, which all together constitute the WSN relational state.

In each iteration, we choose randomly both the sender *s* (node sending information to BS) and a position in the relational space $\mathcal{N}(s)$. It determines the next-hop path forwards to BS (the receiver node) and type of relation describing this hop. In the following, for chosen receiver *r*, we will make a decision taking into account relational state for *sender-receiver* interaction (cells $\pi_{s,r}$, $\vartheta_{s,r}$, $\chi_{s,r}$). Accordingly to obtained results we modify the intensity quotients for *sender-receiver* interaction and repeat hops until information reaches the base station.

Modelling WSN lifetime activity we modify repeatedly matrices π, ϑ, χ, but diagonal elements remains the same. It is apparent that the spatial distribution of intensity quotients fulfils global requirements, but is has captured the essential local relationships within any neighbourhood.

Described above method meets other important requirements for immunity and adaptability. The interaction space within $\mathcal{N}(k)$ shall be modified only in the local vicinity of each node *k* and that all the modified interaction spaces shall then resemble the prevailing activity better then before. They tend to become more similar mutually i.e. differences between any interaction spaces within $\mathcal{N}(C)$ (where $k \in C$) are smoothed.

5. CONCLUSIONS

Dependability of a complex distributed system closely depends on its immunity and adaptation abilities for data exchange. Immunity and adaptation of communication channels are essential in our approach.

We propose a novel relational method which allows reconciling two often dichotomous points of view: immunity and adaptability to neighbourhood. The global network strategy provides each node with an adequate level of immunity and adaptation functions thus guaranteeing sufficient communication services. On the other hand, nodes using these resources provide suitable level of the WSN adaptability and immunity towards the desired common network security level. Such management of complex system yields in growing both system adaptability and immunity.

By modelling WSN activities using relational approach we have managed to accurately describe the complex characteristics of the network interactions, and at the same time we have eliminated many different and distributed variables (the WSN parameters). This reduction can be crucial for system simulation. Hence, with the presented relational approach for modelling immunity functions we are able to scale the complexity of interactions and model with much higher precision various behavioural aspects of WSNs.

REFERENCES

Cohn, A.G., Bennett B., Gooday J.M., Gotts N.M., 1997. Representing and Reasoning with Qalitative Spatial Relations about Regions. In: Cohn, A.G., Bennett B., Gooday J.M., Gotts N.M, eds. *Spatial and Temporal Reasoning*, Dordrecht, Kulwer, 97-134.

Dasgupta D., Immunity-based Intrusion Detection System: A General Framework. Proceedings of the Conference in *22th International Information System Security Conference,* 18-21, Oct. 1999.

Hofmeyr, S., Forrest, S. 2000. Architecture for an Artificial Immune System. *Evolutionary Computation,* vol.8, no.4: 443-473.

Jaron J., 1978. Systemic Prolegomena to Theoretical Cybernetics, *Scientific. Papers of Institute of Technical Cybernetics*, no. 45, Wroclaw University of Technology.

Jungwon, K.,Bentley, P. 2001. Towards an artificial immune system for network intrusiondetection: an investigation of clonal selection with a negativeselection operator, Proceedings of the Conference in *2001 Congress on Evolutionary Computation,* vol.2, 1244-1252. 27-30 May 2001, Seoul, South Korea

Kohonen T., 2001. *Self Organizing Maps*, Springer Series in Information Science, no. 30, Springer-Verlag, Berlin-New York.

Leandro N. de Castro and Jon Timmis. 2002. *Artificial Immune Systems: A new computational intelligence approach*, Springer-Verlag, Great Britain.

Luther, K., Bye, R., Alpcan, T., Muller, A., Albayrak, S. A. 2007. Cooperative AIS Framework for Intrusion Detection, Proceedings of the Conference in *IEEE International Conference on Communications ICC'07*, 1409 -1416. 24-27 June 2007. Glasgow, Scotland.

Nikodem, J., 2008. Autonomy and Cooperation as Factors of Dependability in Wireless Sensor Network, Proceedings of the Conference in *Dependability of Computer Systems, DepCoS - RELCOMEX 2008*, 406-413. 26-28 June 2008, Szklarska Poreba, Poland.

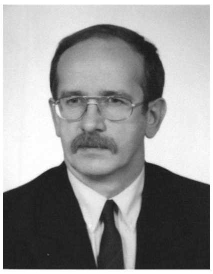
Su P., Feng D., 2006. The Design of an Artificial Immune System, Proceedings of the Conference in *Networking, Systems, Mobile Communications and Learning Technologies (ICNICONSMCL'06)*,

Timmis, J. I.,2000. *Artificial Immune Systems: A novel data analysis technique inspired by the immune network theory*. Ph.D. thesis. Department of Computer Science, University of Wales.

Thomas S., 2006. *On the Appropriateness of Negative Selection for Anomaly Detection and Network*

Intrusion Detection. Ph.D. thesis. Darmstadt University of Technology.

AUTHORS BIOGRAPHY



Jan Nikodem received the B.Sc. in electrical engineering, M.Sc. in artificial intelligence in 1979 and Ph.D. degree in computer science in 1982 from Wroclaw University of Technology (WUT), Poland. Since 1986, he has been an Assistant Professor in the Institute of Technical Cybernetics, WUT. Since 2005 in the Institute of Computer Engineering, Automatics and Robotics (ICEAR). His current research are focused on the area of complex and distributed systems, cybernetics, wireless sensor networks and digital data transmission.

Ryszard Klempous holds a M.Sc. in Automation (1971) and Ph.D. in Computer Science (1980) from Wroclaw University of Technology (WUT). Since 1980 he has been an Assistant Professor in the Institute of Computer Engineering, Automatics and Robotics, WUT. Senior member of IEEE and NYAS, has already published over 90 papers in Optimization Methods and Algorithms, Simulation and Data Processing and Transmission.

Zenon Chaczko completed a B.Sc. in Cybernetics and Informatics in 1980 and a M.Sc. in Economics in 1981 at the University of Economics, Wroclaw in Poland., as well as completed MEng in Control Engineering at the NSWIT 1986, Australia. For over 20 years Mr Chaczko has worked on Sonar and Radar Systems, Simulators, Systems Architecture, Telecommunication network management systems, large distributed Real-Time system architectures, network protocols and system software middleware. Mr Chaczko is a Senior Lecturer in the Information and Communication Group within the Faculty of Engineering at UTS.

TWO-PHASE SIMULATION OPTIMISATION PROCEDURE WITH APPLICATIONS TO MULTI-ECHELON CYCLIC PLANNING

Galina Merkuryeva[a] and Liana Napalkova[b]

Riga Technical University
Department of Modeling and Simulation
1 Kalku Street
LV-1658, Riga, Latvia

[a]gm@itl.rtu.lv, [b]liana@itl.rtu.lv

ABSTRACT

This paper describes a two-phase simulation-based optimisation procedure that integrates the Genetic Algorithm and Response Surface-based Linear Search algorithm for developing optimal power-of-two replenishment policy in multi-echelon environment during the maturity phase of the life cycle of a product. The problem involves a search in high dimensional space with different ranges for decision variables scales, multiple objective functions and problem specific constraints, such as power-of-two and nested/inverted-nested planning policies. The paper provides illustrative example of the two-phase optimisation procedure applied to generic supply chain network.

Keywords: multi-echelon cyclic planning, genetic algorithm, response surface-based linear search

1. INTRODUCTION

For the last years, there has been increasing attention placed on the performance, design, and analysis of multi-echelon supply chains (Merkuryeva and Napalkova 2007). There are two different approaches used to manage supply chains: *single-echelon* and *multi-echelon* approaches.

The single-echelon approach, which includes continuous review, periodic review, single-unit decomposition, Wagner Within, Silver Meal techniques etc., splits multi-level supply chain into separate stages providing suboptimal solutions.

In contrast to it, the multi-echelon approach, which could be applied to non-cyclic and cyclic policies, considers managing all the echelons in a holistic way and, thus, optimises the global supply chain performance. Compared to non-cyclic policies, which are more preferable from theoretical point of view, *cyclic planning* in multi-echelon environment has more practical benefits, because it provides easy control, reduced administrative costs and safety stocks, and elimination of bullwhip effect (Campbell and Mabert 1991). The main idea of cyclic planning is to use cyclic schedules at each echelon and synchronise them with one another (Merkuryev, Merkuryeva, Desmet and Jacquet-Lagrèze 2007).

Optimisation of multi-echelon cyclic plans refers to the class of multi-objective optimisation problems, which are usually characterized by a large search space of decision variables, conflicting and stochastic objectives etc. (Chen 2003). While there is no a single optimal solution for a number of conflicting objectives, the development of an algorithm, which gives a large number of alternative solutions lying near the Pareto optimal front and tackles the variations of a response generated from the uncertainties in the decision variables and/or parameters, is of great practical value.

Different analytical and mathematical programming methods, such as mixed integer programming, stochastic dynamic programming, network programming, etc., have been developed to define optimal cyclic policies (Campbell and Mabert 1991, Federgruen and Zheng 1993). However, real-world supply chains sometimes cannot expect to get a solution by such methods.

The motivation for current study is to propose a two-phase simulation-based optimisation procedure aimed to find optimal parameters of a multi-echelon cyclic policy for each of supply chain nodes, i.e. replenishment cycles and order-up-to levels, during the maturity phase of the life cycle of a product in order to minimize the sum of ordering, production and inventory costs, respecting fill rate's and cyclic planning constraints, taking into account assumptions of stochastic demand, capacity restriction and backorders.

2. OPTIMISATION PROBLEM STATEMENT

The multi-objective simulation-optimisation problem can be symbolically represented in compact form as (Napalkova and Merkuryeva 2008):

$$\text{Min } \mathrm{E}[F(\mathbf{x})] = \mathrm{E}[f_1(\mathbf{x}),\ldots,f_M(\mathbf{x})], \quad (1)$$
$$\text{subject to: } \mathbf{g}(\mathbf{x}) = \mathrm{E}[\mathbf{r}(\mathbf{x})] \le 0 \text{ and } \mathbf{h}(\mathbf{x}) \le 0,$$

where $E[\cdot]$ is a mathematical expectation; $\mathbf{x} = (x_1,\ldots x_K) \in X$, $\mathbf{f} = (f_1,\ldots,f_M) \in F$; K is a number of decision variables; M is a number of objective functions; X is called the decision space; F is the objective space; $\mathbf{x}$ is called a vector of decision variables; $\mathbf{f}$ is a vector of objective functions; $\mathbf{g}$ is a vector of stochastic constraints; $\mathbf{h}$ is a vector of deterministic constraints on the decision variables; $\mathbf{r}$ is a random vector that represents several responses of the simulation model for a given $\mathbf{x}$.

Proceeding from (1), the solution of multi-objective optimisation problem is a vector of decision variables $\mathbf{x}$ that satisfies all feasible constraints and provides the best trade-off between multiple objectives. To describe objective vector function (1), one could use traditional methods of aggregating multiple objectives into a single objective. The main strength of this approach is a computational efficiency and simple implementation. The weakness is the difficulty to determine a value of the weights that reflect a relative importance of each criterion (Abraham, Jain and Goldberg 2005). Therefore, this paper applies the Pareto dominance concept for finding trade-off solutions.

The trade-off solution $\mathbf{x}^* \in F$ is said to be Pareto optimal (or non-dominated) if there does not exist another $\mathbf{x} \in F$ such that $f_i(\mathbf{x}) \le f_i(\mathbf{x}^*)$ for all criterions $i = 1,\ldots,M$ and $f_j(\mathbf{x}) < f_j(\mathbf{x}^*)$ for at least one criterion j (Abraham, Jain and Goldberg 2005). Finding the Pareto optimal set is a necessary condition for selecting trade-off solutions. Note that this definition of Pareto optimality assumes all the objective functions to be minimized. If some objective function f_i is to be maximized, it is equivalent to minimize the function $-f_i$.

Regarding the problem of cyclic planning within multi-echelon supply chain environment we deal with two objective functions. The first one is to minimize the average total cost represented by sum of inventory holding, production and ordering costs in accordance with the following equation:

$$\text{Min E}[TC] = \sum_{t=1}^{T} \Big(\sum_{j=1}^{J} CP_{jt} + \sum_{i=1}^{I} CO_{it} + \sum_{i=1}^{I} CH_{it} \Big), \qquad (2)$$

where TC denotes the total cost, CP_{jt} denotes production cost in process j per period t, CO_{it} is ordering cost at stock point i per period t, and CH_{it} is inventory holding cost at stock point i per period t; I and J correspond to the number of stock points and processes, and T defines the number of periods in the planning horizon.

The second objective function is to maximise customer service requirements specified by the order fill rate.

$$\text{Min E}[FR] = \sum_{t=1}^{T} \sum_{i=1}^{I} \sum_{k=1}^{K} QC_{ikt} \Big/ \sum_{t=1}^{T} \sum_{i=1}^{I} \sum_{k=1}^{K} D_{kit}\ , \qquad (3)$$

where QC_{ikt} is a fraction of orders provided by stock point i to end-customer k in time period t, D_{kit} is actual demand of end-customer k to stock point i in time period t.

Controlled in optimisation experiments, this performance measure is introduced to avoid unconstrained minimization of the total cost.

The decision variables are replenishment cycles and order-up-to levels, which are considered as discrete and continuous type variables, respectively.

The proposed approach is to integrate (a) the Genetic Algorithm (GA) to guide the search towards an approximate Pareto optimal front and (b) Response Surface-based Linear Search (RSLS) algorithm to improve solutions found by GA. The motivation for selecting these algorithms is the following. As it was mentioned above, the optimisation problem (1) is multi-objective and includes both discrete and continuous type variables. GA is well suited for solving multi-objective combinatorial optimisation problems. RSLS is appropriate to improve existing solutions as it is based on local search approach.

3. THE TWO-PHASE OPTIMISATION PROCEDURE

3.1. Conceptual model

The simulation-based optimisation procedure (Merkuryeva, Napalkova and Hatem 2008) developed to create optimal cyclic schedules in multi-echelon environment consists of the following two phases:

- Phase 1: Optimisation of replenishment cycles by Multi-Objective Genetic Algorithm;
- Phase 2: Optimisation of order-up-to levels by Response Surface-based Linear Search.

3.2. Phase 1: Multi-Objective Simulation-based Genetic Algorithm

The Multi-Objective Simulation-based Genetic Algorithm (MOSGA) is aimed to optimising lengths of replenishment cycles in multi-echelon supply chain networks during the maturity phase of the product life cycle. In this phase, corresponding order-up-to levels are calculated using approximate analytical formulas. Table 1 represents main blocks of MOSGA, which are described below. Flowchart of the proposed algorithm is given in Figure 1.

Table 1: Main blocks of MOSGA

Nr	Block	Description
1	Encoding mechanism	Modified binary encoding
2	Initial population	Random initialisation
3	Fitness assignment	Pareto-based ranking
4	Fitness estimation	Through simulation
5	Penalty function	Artificial increase of total cost
6	Crossover & mutation operators	Uniform crossover One-point mutation
7	Diversity preserving mechanism	Crowding distance
8	Selection strategy	Crowded tourn. selection
9	Elitism strategy	$(\mu + \lambda)$ - selection
10	Termination criterion	Fixed nr of generations

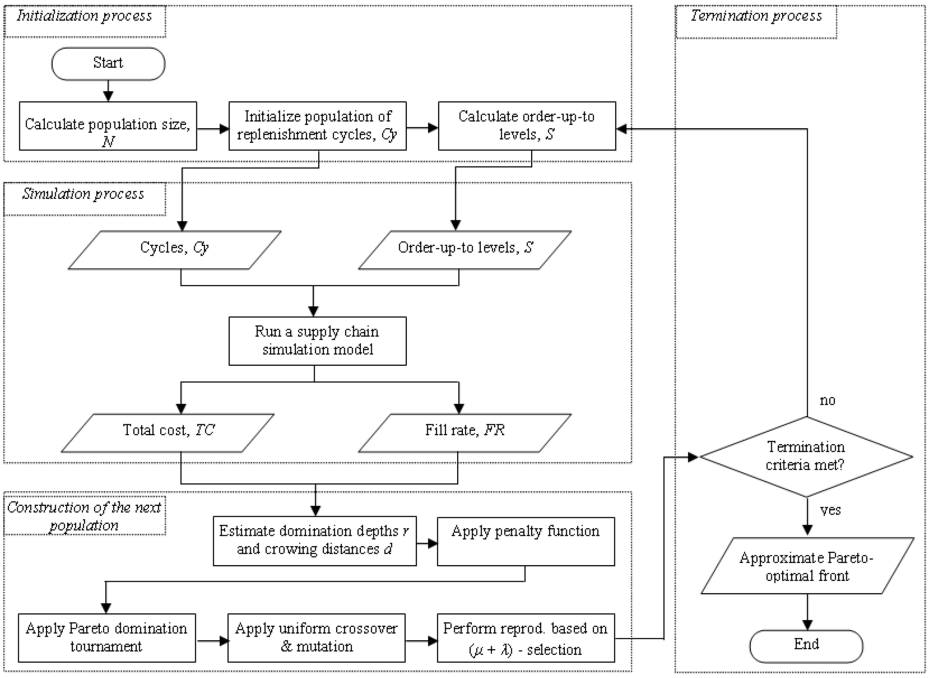

Initialization process
Start
Calculate population size, N
Initialize population of replenishment cycles, Cy
Calculate order-up-to levels, S
Simulation process
Cycles, Cy
Order-up-to levels, S
Run a supply chain simulation model
Total cost, TC
Fill rate, FR
Construction of the next population
Estimate domination depths r and crowing distances d
Apply penalty function
Apply Pareto domination tournament
Apply uniform crossover & mutation
Perform reprod. based on (μ + λ) - selection
Termination process
no
Termination criteria met?
yes
Approximate Pareto-optimal front
End

Figure 1: The MOSGA scheme

3.2.1. Encoding mechanism

Encoding mechanism is aimed to codify replenishment cycles for supply chain stock points that have to satisfy power-of-two policy constraints. Here, the cycles are represented by multiples of two. Therefore, the following modified encoding procedure is introduced that allows codifying cycles by the only powers of the base 2.

For stock point i in chromosome *n*:

1. Represent the cycle Cy_{in} as a multiple of two, i.e. 2^p, $Cy_{in} = t*2^p$, where t is a time-step parameter, p is a power of the base 2.
2. Encode a power p to a binary string a_{Ln}, where L is a length of a chromosome.

For instance, the cycle $Cy_{in} = 28$ could be represented as $7*2^2$, where the time-step parameter $t = 7$ days. Then, the power $p = 2$ is encoded to a binary string $a_{2n} = <1, 0>$, i.e. $0*2^0 + 1*2^1 = 2$.

3.2.2. Initial population

Initial population is generated randomly by using uniform distribution, in order to cover the investigated search space. To define a lower bound of the population size N that guarantees both adequate genetic diversity and reasonable simulation processing time, the following D. Goldberg's formula is used:

$$N = 1.65*2^{(0.2*L)} \qquad (4)$$

The described below steps are applied to create an initial population.

1. Generate a population $P_N = \{Cy_{11} = 2^{p1}, Cy_{21} = 2^{p2}, \ldots, Cy_{IN} = 2^{pN}\}$, where I is a number of stock points in the supply chain, by using "Minimal standard" pseudo-random number generator of Park and Miller.
2. (optional). Sort replenishment cycles subject to nested or inverted-nested policy.
3. Calculate order-up-to levels S_{in} for each stock point i in each chromosome n from the population P_N by using the sequence of analytical approximate formulas described in Napalkova and Merkuryeva (2008).

3.2.3. Fitness assignment and estimation

Fitness is defined based on two objectives represented by performance measures, i.e. the total cost and fill rate that are obtained from simulation experiments. To estimate fitness values of chromosomes, the Pareto-based ranking originally proposed in the *Non-dominated Sorting Genetic Algorithm II* (Deb, Agrawal and Meyariv 2000) is applied.

The example provided below illustrates the concept of the Pareto-based ranking.

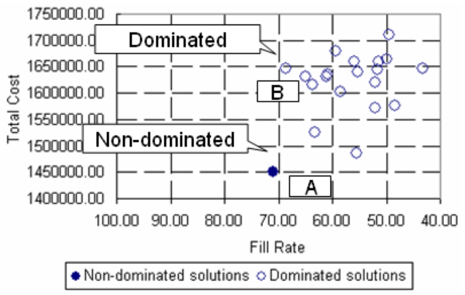


Figure 2: Example of the Pareto-based ranking

The solution represented by the point *A* in Figure 2 is better than the solution in the point *B*, as it gives higher fill rate at lower total cost. This means that the solution *A* is non-dominated and belongs to the first non-dominated front (domination depth is 1), while the solution B is dominated.

The following steps are used for the Pareto-based fitness assignment:

1. Find non-dominated solutions in the entire population and assign them domination depth $r_n = 1$, $n \in R_r$, where R_r is the approximate Pareto optimal set of non-dominated solutions with a domination depth r_n.
2. Temporally exclude non-dominated solutions from the population.
3. Find new non-dominated solutions in the remaining population and assign them domination depth $r_n = r_n + 1$, $n \in R_r$.
4. Repeat Steps 2-3 until all chromosomes are ranked.
5. Reorder solutions according to their fitness values in the increasing sequence, $FITNS = \{fitns_1 \leq fitns_2) \leq \ldots \leq fitns_N\}$.

3.2.4. Penalty function

Penalty function is introduced to decrease the survival probability of solutions, which provide a fill rate lower than the pre-defined threshold. If a solution has a fill rate lower than threshold, its total cost is artificially increased as follows:

$$F_p: TC*k, \qquad (5)$$

where k is a multiplier coefficient.

Here, the fill rate's threshold is defined by 75% and the multiplier coefficient k is equal to 2. In this case, chromosomes with the fill rate below the threshold 75 % are automatically excluded from the population. The value of coefficient k could be also adjusted within the optimisation process.

3.2.5. Crossover and mutation

Crossover and mutation operators provide the exploration and exploitation of a search space. The exploration is used to investigate new and unknown areas in a search space. The exploitation is aimed to making use of knowledge acquired by exploration to reach better positions on the search space. In the proposed GA, the uniform crossover and one-point mutation with probabilities [0.5; 0.8] and [0.01; 0.1], respectively, are introduced.

3.2.6. Diversity preserving mechanism

In order to obtain solutions uniformly distributed over the Pareto front, the diversity preserving mechanism based on a crowding distance is implemented in MOSGA.

The crowding distances d_n are calculated based on the values of total cost and fill rate that are normalized in the interval [0.0; 1.0].

1. For each chromosome n initialize crowding distances $d_n = 0$, $n = 1,\ldots,N$.
2. Sort chromosomes a_{Ln}, $n = 1,\ldots,N$ subject to normalised total cost $f_n^{1,norm}$.
3. Assign $d_1 = \infty$.
4. For each crowding distance d_n, $n = 2,\ldots,N\text{-}1$ do Step 5.
5. $d_n = d_n + d^1_{n-1,n+1}$, $d^1_{n-1,n+1} = f_{n+1}^{1,norm} - f_{n-1}^{1,norm}$.
6. Sort chromosomes a_{Ln}, $n = 1,\ldots,N$ subject to normalised fill rate $f_n^{2,norm}$.
7. Assign $d_N = \infty$.
8. For each crowding distance d_n, $n = 2,\ldots,N\text{-}1$ do Step 9.
9. $d_n = d_n + d^2_{n-1,n+1}$, $d^2_{n-1,n+1} = f_{n+1}^{2,norm} - f_{n-1}^{2,norm}$.

The main advantage of the crowding approach is that a user doesn't have to define additional parameters, such as, for example, a sharing parameter.

3.2.7. Selection strategy

Crowded tournament selection originally proposed in NSGA-II [14] is used in MOSGA. The main idea of this selection strategy is that a crowded comparison operator is applied to choose the better chromosome in randomly selected ones. Because domination depth is used to assign fitness values, chromosomes with the same domination depth could be selected. In this case, to define the better chromosome, an additional attribute is introduced. In the algorithm, a crowding distance, which is an estimate of the density of solutions surrounding the current solution, is proposed as the additional attribute. Chromosomes with bigger crowding distances have more chances to be selected. Crowding distances of chromosomes, which provide the best value for each objective, are assigned to 999999 that means ∞. As a result, every chromosome in the population has the following two attributes: (1) domination depth and (2) crowding distance. From two solutions the solution with the lower domination depth is preferable. If both solutions have the same depth then the solution with larger crowding distance is preferable.

The crowded comparison operator ($\geq$) is defined as follows:

$$a_{Lj} \geq a_{Lk} \text{ if } (r_j < r_k) \text{ or } ((r_j = r_k) \text{ and } (d_j < d_k)), \qquad (6)$$

where d_j and d_k are crowding distances for chromosomes j and k.

3.2.8. Elitism strategy

Elitism strategy is used to avoid the loss of non-dominated solutions during the evolution process. In each generation, an offspring population is added to a parents' population. Domination depths of chromosomes in the combined population are updated. First N solutions are gathered for the next generation, where N is a population size. This strategy is often called as $(\mu + \lambda)$ – selection, where μ and λ assign parents and mating pool, respectively.

The Genetic Algorithm automatically stops the optimisation when the pre-defined number of populations is generated.

3.3. Phase 2: Response Surface-based Linear Search algorithm

The algorithm in the phase 2 is aimed to improve the cyclic planning solution of the Genetic Algorithm received in phase 1 by adjusting analytically calculated order-up-to levels of stock points that could result in decreasing the total cost and/or increasing the end-customers fill rate. It is based on the Response Surface-based Methodology (RSM) applied to simulation optimisation problem (Merkuryeva 2005).

The developed Response Surface-based Linear Search (RSLS) algorithm is based on local approximation of the simulation response surface by a regression type meta-model in a small region of independent factors and integrates linear search techniques for optimising stock points' order-up-to levels. A linear search is a sequential procedure that in each iteration *m* includes the following main building blocks:

1. Local approximation of the response surface function,
2. Checking the fit of a meta-model,
3. Linear search in the steepest descent direction,
4. Updating the Pareto front.

3.3.1. Local approximation of the response surface function

The total cost is defined as a simulation response and order-up-to levels as input factors. Local approximation of the response surface function in the small region of input factors is performed by using a first-order model. The small region of input factors, or a local search space, is described by a central point, lower and upper bounds.

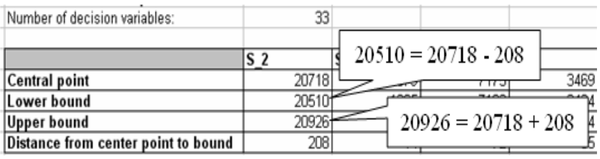
In iteration *m* lower t_i^m and upper u_i^m bounds of the local search space, or a region of experimentation, are defined as *proc* % decrease and increase from the central point, respectively:

$$t_i^m = \xi_{i0}^m - proc * \xi_{i0}^m \, , \; i = 1,...,I \, , \tag{7}$$

$$u_i^m = \xi_{i0}^m + proc * \xi_{i0}^m \, , \; i = 1,...,I \, , \tag{8}$$

where $\xi^m{}_{i0}$ is a central point of input factor *i*.

An example of a local search space definition in a set of order-up-to levels is given in Figure 3.



Number of decision variables:	33			
	S_2	S		
Central point	20718	[illegible]	[illegible]	3469
Lower bound	20510	[illegible]	[illegible]	[illegible]
Upper bound	20926	[illegible]	[illegible]	[illegible]
Distance from center point to bound	208	[illegible]	[illegible]	[illegible]

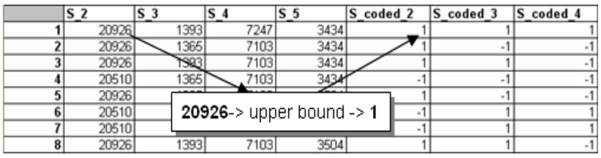
Figure 3: Example of local search space calculation

To increase the numerical accuracy in estimation of regression coefficients, input factors are coded by the following formula:

$$x_i^m = \frac{\xi_i^m - \xi_{i0}^m}{c_i^m} \, , \; i = 1,...,I \, , \tag{9}$$

where x_i^m is a coded input factor *i*; c_i^m is a distance between the central point and lower (or upper) bound.

Encoding procedures for order-up-to levels are illustrated in Figure 4.



	S_2	S_3	S_4	S_5	S_coded_2	S_coded_3	S_coded_4
1	20926	1393	7247	3434	1	1	1
2	20926	1365	7103	3434	1	-1	-1
3	20926	1393	7103	3434	1	1	-1
4	20510	1365	7103	3434	-1	-1	-1
5	20926	[illegible]	[illegible]	[illegible]	1	-1	-1
6	20510	[illegible]	[illegible]	[illegible]	-1	1	1
7	20510	[illegible]	[illegible]	[illegible]	-1	1	1
8	20926	1393	7103	3504	1	1	-1

Figure 4: Example of encoding procedure

To make local approximation of a simulation response surface function, the first-order regression meta-model for coded input factors that describes main effects of input factors is used in iteration *m*:

$$y = b_0^m + \sum_{i=1}^{q} b_i^m x_i^m + \varepsilon^m \, , \; i = 1,...,I \, , \tag{10}$$

where b_i^m is a regression coefficient of input factor *i*; ε^m is a statistical error of a regression model.

In order to estimate meta-model coefficients b_i^m from simulation experiments, the Plackett-Burman experimental design added by simulation replications in the central point is applied. The template for the proposed design has been generated in Minitab statistical software.

Finally, estimates of coefficients of the meta-model (10) are calculated by using well-known method of least squares.

3.3.2. Checking the fit of a meta-model

Lack-of-Fit test is performed to check the adequacy of a regression meta-model and to verify the least-squares method assumptions. Testing lack-of-fit, that gives the determination coefficient close to 1 (100%) and p-value < 0.05 implies the resulted meta-model to be adequate.

3.3.3. Linear search in the steepest descent direction

A linear search is performed within the local search space for order-up-to levels, in the steepest descent direction defined by a vector (b_1^m, b_2^m,…, b_q^m) starting from the central point, where b_1^m, b_2^m,…, b_q^m are coefficients of the simulation meta-model received in iteration *m*. The increment along the projection of the search direction is calculated only if corresponding regression coefficient is significant (p-value < 0.05). These increments of coded Δx_i^m and decoded Δ_i^m input factor *i* are calculated as follows:

$$\Delta x_i^m = \frac{-b_i^m}{\max\limits_i \left| b_i^m \right|} \, , \; i = 1,...,I \, , \tag{11}$$

$$\Delta_i^m = \Delta x_i^m * c_i^m, \; i = 1,\dots,I . \tag{12}$$

The values of decoded input factors in the next step of a linear search within iteration *m* are calculated as follows:

$$\xi_i^m = \xi_i^m + \Delta_i^m, \; i = 1,\dots,I . \tag{13}$$

The following termination rules are proposed to stop the linear search in iteration *m*:

1. The simulation response value could not be improved,
2. The search goes outside the pre-defined local search space.

Lower and upper bounds of a new experimental region in iteration *m+1* are updated by using formulas (7, 8).

3.3.4. Updating the Pareto front

Solutions found during the RSLS are included in the approximate Pareto optimal set initially generated by the Genetic Algorithm. Then all solutions in the resulting Pareto optimal set are reordered according to their fitness values in the increasing sequence.

4. RESULTS AND ANALYSIS

4.1. User interface

The user-interface of MOSGA is developed in MS Excel using ActiveX controls. There are three main user-interface windows, which are used to group the input data, GA and simulation options.

The input data window defines the number of input factors, minimal and maximal values of replenishment cycles, synchronization policy, etc.

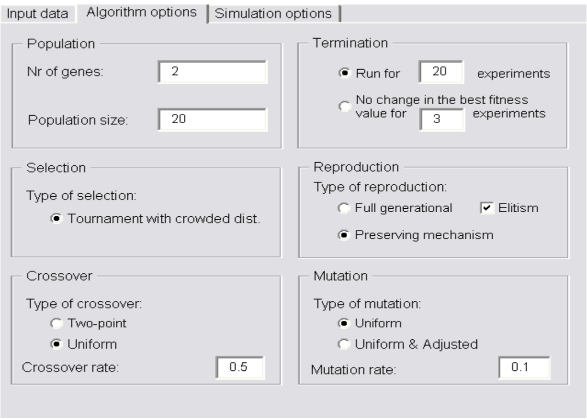
The window of algorithm options (Figure 5) describes the number of genes to encode input factors, population size, a type of selection and reproduction strategies, etc.

The window of simulation options describes the number of simulation replication per simulation experiment, length of simulation run warm-up period in hours.

Control buttons allow the user to load the simulation model, run and terminate the optimisation algorithm, as well as calculate the population size based on D. Goldberg's formula (4) for binary-encoded chromosomes in such a way that every solution in the search space is attainable with the crossover genetic operator.

4.2. Input data description

The application itself is aimed to find an optimal cyclic plan of a chemical product, i.e. liquid based raisin, in order to minimise inventory holding, ordering and production costs, and maximise end-customers fill rate. As a test bed, the chemical manufacturing supply chain is used. The main operations occurred in the supply chain network are the following. In the plant CH (see, Figure 6), the raw material is converted to the liquid based raisin. It is then either sourced to direct customers or shipped to the plant DE, where other components are added to make different products. From that plant, the end-products are shipped to different types of customers.



Population
Nr of genes: 2
Population size: 20

Termination
● Run for 20 experiments
○ No change in the best fitness value for 3 experiments

Selection
Type of selection:
● Tournament with crowded dist.

Reproduction
Type of reproduction:
○ Full generational
☑ Elitism
● Preserving mechanism

Crossover
Type of crossover:
○ Two-point
● Uniform
Crossover rate: 0.5

Mutation
Type of mutation:
● Uniform
○ Uniform & Adjusted
Mutation rate: 0.1

Figure 5: Example of MOSGA user-interface

The ProModel-based simulation model of the above-described supply chain network is generated automatically using a simulation-based optimisation environment presented in Merkuryeva and Napalkova 2007. The end-customer demand is normally distributed; and replenishment cycles are defined according to the power-of-two policy. Cycles are presented in weeks as follows, 7, 14, 28, 56, where 56 days is the maximal cycle that corresponds to one full turn of a "planning wheel". In this business case, specific policies such as nested or inverted-nested ones are not analysed. Initial stocks are equal to order-up-to levels plus average demand multiplied by cycle delays. Stock point 1 has infinite on hand stock and is not controlled by any policy. Backorders are delivered in full.

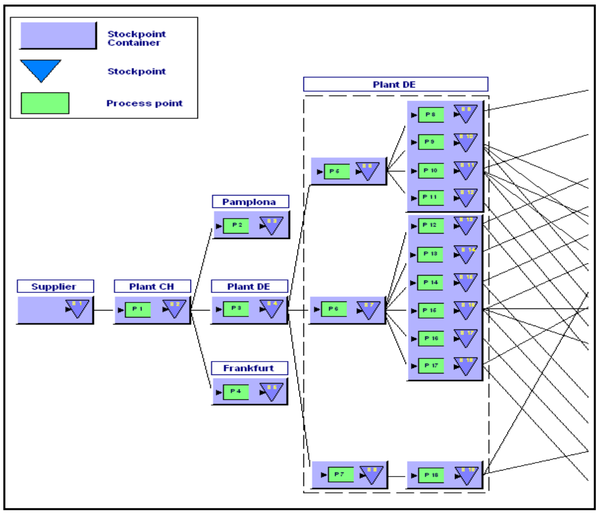


Figure 6: Example of a generic network simulation model

Simulation run length T is equal to 224 periods. The warm-up period is defined by 112 periods. This allows modelling of two full turns of the "planning wheel", i.e. 2*56 periods, during the warm-up period. Number of simulation replications is equal to 5.

4.3. Solutions found by the Genetic Algorithm

To optimise replenishment cycles, GA is executed with parameters summarized in Table 2.

Table 2: Parameters of the Genetic Algorithm

Parameter	Value
Population size	20
Crossover probability	0.5
Mutation probability	0.1
Tournament size	2

The algorithm works with 33 decision variables, which are assigned to network stock points. When the number of generated population is equal to 16, the algorithm is terminated.

Figures 2 and 7 show solutions received from initial and final populations. It could be noticed that at the beginning the approximate Pareto optimal set includes a single non-dominated solution. However, during the evolution process the diversity of the approximate Pareto optimal set is increased, and the final population includes seven non-dominated solutions (Figure 7).

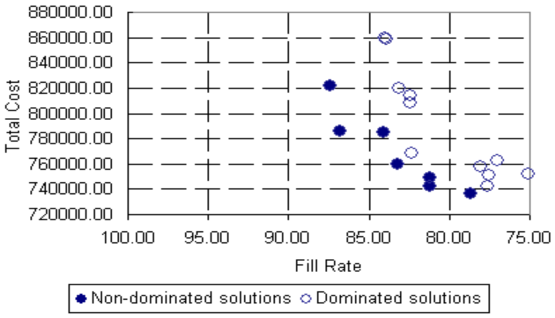


Figure 7: Final population mapped in the objective space

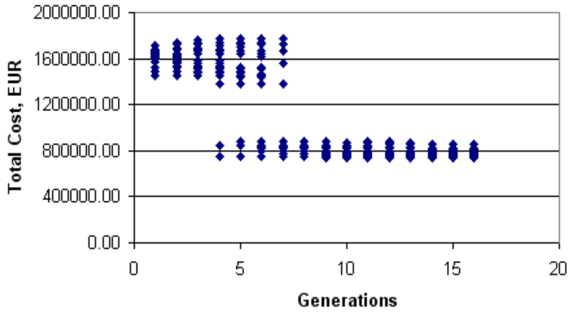


Figure 8: The Genetic Algorithm's convergence subject to Total Cost

Figures 8 and 9 illustrate execution of the Genetic Algorithm. The total cost and fill rate of parent chromosomes are plotted against the generation step. The algorithm makes quick progress in the beginning of the evolution that is typical for Genetic Algorithms. Then, there are phases when it hits the local optimum before mutations further improve its performance.

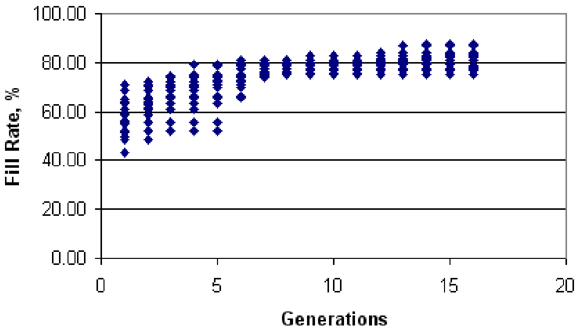


Figure 9: The Genetic Algorithm's convergence subject to Fill Rate

4.4. Solutions adjusted by the Response Surface-based Linear Search algorithm

For each non-dominated solution received in phase 1, stock points' replenishment cycles are fixed and their order-up-to levels are optimised by RSMS algorithm.

Next, the Pareto front generated by GA is updated by adding solutions found within the RSMS (Figure 10).

As a result, we get that these solutions dominate the solutions received in the phase 1 and could substitute them in the final approximate Pareto front (Figure 11).

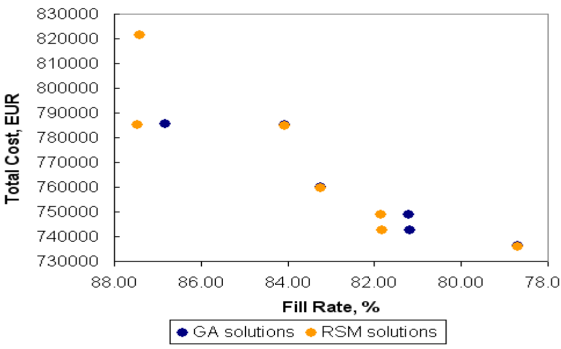


Figure 10: Solutions of MOSGA and RSMS algorithms

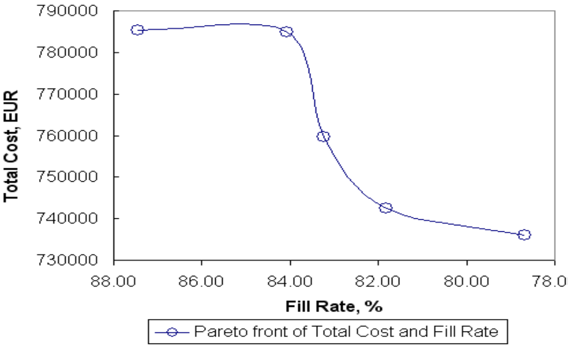


Figure 11: The final approximate Pareto front

5. CONCLUSIONS

The paper presents the simulation-based optimisation algorithm developed in order to define the optimal lengths of cycles and stock point order-up-to levels during the maturity phase of the life cycle of a product. It integrates the Genetic Algorithm used to optimise replenishment cycles and Response Surface-based Linear Search Algorithm that allows adjusting order up-to levels of stock points when replenishment cycles are fixed. The algorithm is applied to optimise parameters of cyclic plans in multi-echelon supply chain. The achieved results show the efficiency of the developed optimisation procedure. Future research will focus on the performance analysis of the developed procedure and algorithms in solving new problem instances.

ACKNOWLEDGMENTS

The presented research is supported by the European Social Fund within the National Programme "Support for the carrying out doctoral study programmes and post-doctoral researches" project "Support for the development of doctoral studies at Riga Technical University".

The authors would like to thank Jonas Hatem from Möbius Ltd. for helpful comments and discussions.

REFERENCES

Abraham, A., Jain, L., Goldberg, R. 2005. *Evolutionary Multiobjective Optimisation: theoretical advances and applications*. United States of America: Springer.

Campbell, G.M., Mabert, V. A., 1991. Cyclical Schedules for Capacitated Lot Sizing with Dynamic Demands. *Management Science*, pp. 409 – 427.

Chen, J.-H., 2003 Theoretical Analysis of Multi-Objective Genetic Algorithms – Convergence Time, Population Sizing and Disequilibrium. *Report for IEEE NNS Walter Karplus Research Grant*.

Deb, K., Agrawal, S., Meyarivan, T., 2000. A Fast Elitist Non-Dominated Sorting Genetic Algorithm for Multi-Objective Optimisation: NSGA-II. *Proceedings of the Parallel Problem Solving from Nature VI Conference*, Springer, Lecture Notes in Computer Science No. 1917, pp. 849 - 858.

Federgruen, A., Zheng, Y-S., 1993. Efficient Algorithms for Finding Optimal Power-Of-Two Policies for Production/Distribution Systems with General Joint Setup Costs. *Operation Research*, Vol. 43, No. 3, pp. 458 – 470.

Merkuryev, Y., Merkuryeva, G., Desmet, B., Jacquet-Lagrèze, E., 2007. Integrating Analytical and Simulation Techniques in Multi-Echelon Cyclic Planning. *Proceedings of the First Asia International Conference on Modelling and Simulation (AMS 2007)*, pp. 460 – 464. March 27 – 30, Phuket, Thailand.

Merkuryeva, G., Napalkova, L., 2007. Development of simulation-based environment for multi-echelon cyclic planning and optimisation. *Proceedings of 6th EUROSIM Congress on Modelling and Simulation*, paper ID 452. Ljubljana, Slovenia.

Merkuryeva, G., Napalkova, L., Hatem, J., 2008. *Deliverable D2.1.5: Report on the response surface based meta-modelling optimisation algorithm for defining lengths of cycles during maturity phase of the life cycle of a product*. Eclips project.

Merkuryeva, G., 2005. Response surface-based simulation metamodelling methods with applications to optimisation problems. Chapter 15. *Supply chain optimisation Product / Process Design, Facility Location and Flow control* / Eds. A. Dolgui, J. Soldek and O.Zaikin, Springer, pp. 205 – 215.

Napalkova, L., Merkuryeva, G., 2008. Theoretical Framework of Multi-Objective Simulation-Based Genetic Algorithm for Supply Chain Cyclic Planning and Optimisation. *Proceedings of 10th International Conference of Computer Modelling and Simulation.* Cambridge, England.

Veldhuizen, V. A., 1999. *Multi-objective Evolutionary Algorithms: Classifications, Analysis and New Innovations*. Thesis (PhD). Air Force Institute of Technology, Wright-Patterson AFB, Ohio.

AUTHORS BIOGRAPHY

GALINA MERKURYEVA is Professor at the Department of Modelling and Simulation, Riga Technical University. She holds two degrees: DSc and Dr.sc.ing. G. Merkuryeva has professional interests and experiences in discrete-event simulation, simulation metamodeling and optimisation, simulation-based training, and supply chain simulation and decision support.

LIANA NAPALKOVA holds her MSc degree in Computer Science from Riga Technical University (2006). Currently, she is a PhD student at RTU Department of Modelling and Simulation, and participates in research projects in the logistics field. Her interests focus on the use of simulation-based optimisation techniques for providing competitive advantages in multi-echelon supply chain cyclic planning. Liana Napalkova is a member of the Latvian Simulation Society.

EFFECTIVE ALLELE PRESERVATION BY OFFSPRING SELECTION: AN EMPIRICAL STUDY FOR THE TSP

Michael Affenzeller[a], Stefan Wagner[b], Stephan Winkler[c]

[a],[b],[c] Upper Austrian University of Applied Sciences,
School of Informatics, Communications and Media
Heuristic and Evolutionary Algorithms Laboratory
Softwarepark 11, 4232 Hagenberg, Austria

[a]michael.affenzeller@heuristiclab.com, [b]stefan.wagner@heuristiclab.com, [c]stephan.winkler@heuristiclab.com

ABSTRACT

The basic selection ideas of the different representatives of evolutionary algorithms are sometimes quite diverse. The selection concept of genetic algorithms (GAs) and genetic programming (GP) is basically realized by the selection of above-average parents for reproduction whereas evolution strategies (ES) use the fitness of newly evolved offspring as the basis for selection (survival of the fittest due to birth surplus). This contribution considers aspects of population genetics and Evolution Strategies in order to propose an enhanced and generic selection model for Genetic Algorithms which is able to preserve the alleles which are part of a high quality solution. Some selected aspects of these enhanced techniques are discussed exemplarily on the basis of travelling salesman benchmark (TSP) benchmark problem instances.

Keywords: softcomputing, evolutionary computation, selection, self adaptation

1. INTRODUCTION

As some kind of approximation for the gradient information which is not available for many optimization problems, neighborhood search aims to obtain information about the descent/increase of the objective function in the local neighborhood of a certain point. Conventional neighborhood search starts from an arbitrary point in the search space and iteratively moves to more and more promising points along a given neighborhood structure (w.r.t. the objective function) as long as no better solution can be detected in the local neighborhood.

The self-evident drawback of this method is that for more complex functions the algorithm converges and gets stuck in the next attracting local optimum which may be far away of a global optimum. It is a common feature of all methods based upon neighborhood search to counteract this essential handicap.

Simulated annealing (SA) for example also allows moves to worse neighborhood solutions with a certain probability which decreases as the search process progresses in order to scan the solution space broader at the beginning and to become more and more goal-oriented at the end. Tabu search on the other hand introduces some kind of memory in terms of a so-called tabu list which stores moves that are considered to lead to already visited areas of the search space. However, also evolution Strategies (ES), a well-known representative of Evolutionary Computation, have to be considered as some kind of parallel neighborhood search as asexual mutation (a local operator) is the only way to create new individuals (solution candidates) in the classical ES-versions. Therefore, in the case of multimodal search spaces, global optima are detected by an ES only if one of the starting values is located in the absorbing region (attracting basin) of a global optimum.

By considering recombination (crossover) as their main operator, GAs and also GP take a basically different approach compared to neighborhood-based techniques as recombination is a sexual operator, i.e. properties of individuals from different regions of the search space are combined in new individuals. Therefore, the advantage of applying GAs to hard problems of combinatorial optimization lies in their ability to scan broader regions of the solution space than heuristic methods based upon neighborhood search do.
Nevertheless, also GAs are frequently faced with a problem which, at least in its impact, is quite similar to the problem of stagnating in a local but not global optimum. This drawback, called premature convergence in the terminology of GAs, occurs if the population of a Genetic Algorithm reaches such a suboptimal state that the genetic operators are no longer able to produce offspring that are able to outperform their parents (Fogel 1996; Affenzeller 2005). This happens if the genetic information stored in the individuals of a population does not contain that genetic information which would be necessary to further improve the solution quality. Therefore, in contrast to the present contribution, the topic of premature convergence is considered to be closely related to the loss of genetic variation in the entire population in GA-research.

In this contribution we do not identify the reasons for premature convergence in the loss of genetic variation in general but more specifically in the loss of what we say essential genetic information, i.e. in the loss of alleles which are part of a global optimal solution. Therefore, we will denote the genetic information of the global optimal solution (which is usually unknown a priori) as essential genetic information in the following.

But what are the reasons for premature convergence, or in other words what are the reasons that this essential genetic information is not or no more available:

- Firstly, one reason for this loss of essential genetic information may be that these alleles are simply not represented in the initial population of the Genetic Algorithm.
- Then, especially in the earlier phase of genetic search it frequently happens that essential genetic information is hidden in individuals with bad total fitness and is therefore eliminated due to selection.
- Furthermore, for the majority of GA applications it is absolutely not guaranteed that the applied crossover operators are able to create new children in a way that the newly evolving child contains exactly the genetic information of its own parents. If this is not guaranteed this fact represents a further reason for a genetic algorithm to loose essential genetic information and therefore cause premature convergence.

The main measure in conventional GA-theory to counteract against this phenomenon is mutation (and migration in the parallel variants) and indeed - as will be shown in the empirical part of the paper - this works quite well and a lot of already lost essential genetic information can be recovered by mutation.

The main aim of the present work is to discuss, analyze and improve new generic theoretical concepts for avoiding or at least retarding premature convergence in a non-problem-specific way by taking the above stated considerations into account:

A very essential question about the general performance of a GA is, whether or not good parents are able to produce children of comparable or even better fitness (the building block hypothesis implicitly relies on this). In natural evolution, this is almost always true. For Genetic Algorithms this property is not so easy to guarantee. The disillusioning fact is that the user has to take care of an appropriate coding in order to make this fundamental property hold. In order to somehow overcome this strong requirement we try to get to the bottom of reasons for premature convergence from a technical as well as from a population genetics inspired point of view and draw some essential interconnections.

The basic idea of the new selection model, introduced as offspring selection (Affenzeller and Wagner 2004a) is to consider not only the fitness of the parents in order to produce a child for the ongoing evolutionary process. Additionally, the fitness value of the evenly produced child is compared with the fitness values of its own parents. The child is accepted definitely as a candidate for the further evolutionary process if and only if the reproduction operator was able to produce a child that could outperform the fitness of its own parents. This strategy guarantees that evolution is presumed mainly with crossover results that were able to mix the properties of their parents in an advantageous way. I.e. **survival of the fittest alleles is rather supported than survival of the fittest individuals** which is a very essential aspect for the preservation of essential genetic information stored in many individuals (which may not be the fittest in the sense of individual fitness).

The experimental part analyzes the characteristics of offspring selection on the basis of some rather small TSP benchmark problems: As commonly done when evaluating the capability of heuristic techniques, some main features are analyzed separately.

2. SELECTION

In terms of goal orientedness, selection is the driving force of GAs. In contrast to crossover and mutation, selection is completely generic, i.e. independent of the actually employed problem and its representation. A fitness function assigns a score to each individual in a population that indicates the 'quality' of the solution the individual represents. The fitness function is often given as part of the problem description or based upon the objective function.

In the Standard Genetic Algorithm the probability that a chromosome in the current population is selected for reproduction is proportional to its fitness (roulette wheel selection). However, there are also many other ways of accomplishing selection. These include for example linear-rank selection or tournament selection (Michalewicz 1996; Schöneburg, Heinzmann and Feddersen, 1994).

However, all evenly mentioned GA-selection principles have one thing in common:

They all just consider the aspect of sexual selection, i.e. mechanisms of selection only come into play for the selection of parents for reproduction. The enhanced selection model which will be described in the following section defies this limitation by considering selection in a more general sense.

2.1. Parent Selection vs. Offspring Selection

In the following we describe and aim to bring together technical as well as biologically motivated considerations in order to motivate the concepts proposed in this paper. The following listing itemizes the most essential aspect in the development phase of the new methods in the above mentioned sense

2.1.1. Selection and Selection Pressure

In the theory of Genetic Algorithms selection and selection pressure are predetermined by the so-called mating scheme and by the replacement strategy actually deployed. By that it should be achieved that offspring of highly fit individuals are represented in the next generation with a higher probability than offspring of average or below average individuals. The goal of this procedure is a continuous advancement of the population over the generations. Typical mating schemes are roulette wheel, linear rank or tournament. This classical GA selection concept is known as sexual selection in the terminology of population genetics. In the population genetics view, sexual selection covers only a rather small aspect of selection which appears when individuals have to compete to attract mates for reproduction. The population genetics basic selection model basically considers the selection process in the following way:

Random mating → selection → random mating → selection → ……….

I.e. selection is considered to depend mainly on the probability of surviving of newborn individuals until they reach pubescence which is called viability in the terminology of population genetics.

The essential aspect of offspring selection in the interpretation of selection is rarely considered in conventional GA selection. The classical (μ, λ) Evolution Strategy in contrast does this very well: Reconsidering the basic functioning of a (μ, λ)-ES in terms of selection μ parents produce λ $(\mu \leq \lambda)$ offspring from which the best μ are selected as members of the next generation. In contrast to GAs where selection pressure is predetermined by the choice of the mating scheme and the replacement strategy, ES allow an easy steering of selection pressure by the ratio between λ and μ. The selection pressure steering model introduced in (Affenzeller 2001) and further developed in (Affenzeller and Wagner 2004) picks up this basic idea of ES and transforms these concepts for GAs in order to have an adjustable selection pressure (independent of the mating scheme and replacement strategy) at one's disposal.

Our advanced selection scheme allowing self-adaptive steering of selection pressure aims to transform the basic ideas for improving the performance of GAs. In doing so the survival probability is determined by a comparison of the fitness of the newly generated individual with the fitness values of its parents. Indeed, as demonstrated in the experimental part, it appears that the first sexual selection step (selection before reproduction) as in case of a standard GA does not drastically effect the qualitative or quantitative performance of the algorithm if being equipped with the newly defined offspring selection step (selection after reproduction). Even with random sexual selection (corresponding to the basic model of the population genetic's selection model) the results are about the same or even better than with roulette wheel or linear-rank as the first selection step. A very important consequence of selection in population genetics as well as in evolutionary computation is its influence on certain alleles. As a matter of principle there are four possibilities for each allele in the population:

- The allele may be fixed in the population
- The allele may disappear from the population
- The allele may converge to an equilibrium state
- No change in allele frequency

The basic approaches for retarding premature convergence discussed in GA literature aim to maintain genetic diversity. The most common techniques for this purpose are based upon preselection (Cavicchio1970), crowding (DeJong 1975), or fitness-sharing (Goldberg 1989). The main idea of these techniques is to maintain genetic diversity by the preferred replacement of similar individuals (Cavicchio1970), (DeJong 1975) or by the fitness-sharing of individuals which are located in densely populated regions (Goldberg 1989). While methods based upon crowding (DeJong 1975) or fitness sharing (Goldberg 1989) require some kind of neighborhood measure depending on the problem representation, (Goldberg 1989) is additionally quite restricted to proportional selection. Moreover, these techniques have the common goal to maintain genetic diversity which is very important in natural evolution where a rich gene pool is the guarantor in terms of adaptiveness w.r.t. changing environmental conditions.

In case of artificial genetic search as being performed by a Genetic Algorithm the optimization goal does not change during the run of a GA and the fixing of alleles of high quality solutions is desirable in the same manner as the erasement of alleles which are definitely not part of a good solution in order to reduce the search space and make genetic search more goal-oriented. I.e. we claim that pure diversity maintenance mechanisms as commonly proposed in GA theory do not support goal-oriented genetic search w.r.t the locating of global optimal solutions.

2.1.2. Adjustable Selection Pressure

One interpretation of GA-selection similar to the concepts of a (μ, λ)-ES is to generate an intermediate population (what is called virtual population in our notation) of size $|POP|\cdot T$ (where $T \geq 1$) by sexual selection, crossover and mutation from the actual population of size $|POP|$.

Then, similar to the interpretation of ES-selection, the best |POP| members from the virtual population are chosen as members of the real next generation that contains the genetic information for the evolutionary process yet to come.

The remaining $(1-T)\ |POP|$ candidates can be seen as individuals that do not reach the age of sexual maturity. A practical problem in the technical appliance of this technique is that it does not contain any indicator about the effectiveness of actual genetic search, the effectiveness of the actually used operators, etc.

I.e. there is no information about the amount of selection pressure to be employed at a certain stage of genetic search. The aim is on the one hand to provide enough selection pressure for not losing essential building block information. On the other hand, too much selection pressure may support unwanted premature convergence to a suboptimal solution. Even if this concept of selection pressure steering has already proven to be very powerful in terms of stability and global solution quality (Affenzeller 2001; Affenzeller 2002) it is a time consuming task to find an advantageous steering of (T) that requires an experienced user.

These considerations already highly indicate the need for some kind of self-adaptation. The essential question is how to introduce self-adaptation into the GA-selection process in a generic i.e. non problem specific way. The approach which we have developed for this reason will be described in the following.

3. OFFSPRING SELECTION

The basic idea to create and evaluate a certain amount (greater or equal population size) of offspring, to be considered for future members of the next generation, is adapted from Evolution Strategies. Self-adaption comes into play when considering the question which amount of offspring is necessary to be created at each round, and which of these candidates are to be selected as members of the next generation, i.e. for the ongoing evolutionary process. In order to keep the concepts generic, no problem specific information about the solution space is allowed to be used for stating the self-adaptive model. Thus, it is desirable to systematically utilize just the fitness information of the individuals of the actual generation for building up the next generation of individuals, in order to keep the new concepts and methods generic. In principle, the new selection strategy acts in the following way:

The first selection step chooses the parents for crossover either randomly or in the well-known way of Genetic Algorithms by proportional, linear-rank, or some kind of tournament selection strategy. After having performed crossover and mutation with the selected parents we introduce a further selection mechanism that considers the success of the apparently applied reproduction in order to assure the proceeding of genetic search mainly with successful offspring in that way that the used crossover and mutation operators were able to create a child that surpasses its parents' fitness. Therefore, a new parameter, called success ratio (*SuccRatio* $\in [0, 1]$), is introduced. The success ratio gives the quotient of the next population members that have to be generated by successful mating in relation to the total population size. Our adaptation of Rechenberg's success rule (Rechenberg 1973) for Genetic Algorithms says that a child is successful if its fitness is better than the fitness of its parents, whereby the meaning of 'better' has to be explained in more detail: is a child better than its parents, if it surpasses the fitness of the weaker, the better, or is it in fact some kind of mean value of both?

For this problem we have decided to introduce a cooling strategy similar to Simulated Annealing. Following the basic principle of Simulated Annealing we claim that an offspring only has to surpass the fitness value of the worse parent in order to be considered as 'successful' at the beginning and while evolution proceeds the child has to be better than a fitness value continuously increasing between the fitness of the weaker and the better parent. Like in the case of Simulated Annealing, this strategy effects a broader search at the beginning whereas at the end of the search process this operator acts in a more and more directed way. Having filled up the claimed ratio (*SuccRatio*) of the next generation with successful individuals in the above meaning, the rest of the next generation *((1-SuccRatio)·|POP|)* is simply filled up with individuals randomly chosen from the pool of individuals that were also created by crossover but did not reach the success criterion. The actual selection pressure *ActSelPress* at the end of a single generation is defined by the quotient of individuals that had to be considered until the success ratio was reached and the number of individuals in the population in the following way:

$$ActSelPress = \frac{|POP|SuccRatio + |POOL|}{|POP|}$$

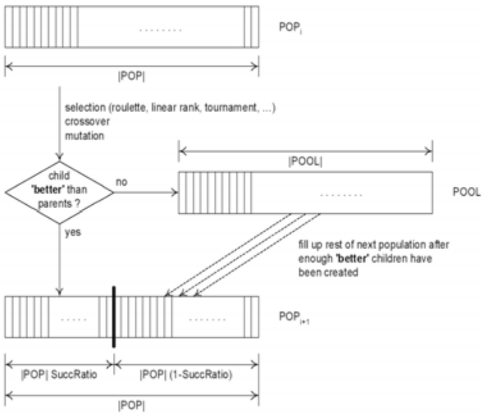

POP_i
|POP|
selection (roulette, linear rank, tournament, ...)
crossover
mutation
|POOL|
child "better" than parents ?
no
POOL
yes
fill up rest of next population after enough "better" children have been created
POP_i+1
|POP| SuccRatio
|POP| (1-SuccRatio)
|POP|

Figure 1: Flowchart for Embedding the new Selection Principle (Offspring Selection) into a Genetic Algorithm

Figure 1 shows the operating sequence of the above described concepts. With an upper limit of selection pressure *MaxSelPress* defining the maximum number of children considered for the next generation (as a multiple of the actual population size) that may be produced in order to fulfill the success ratio, this new model also functions as a precise detector of premature convergence:

> *If it is no longer possible to find a sufficient number of SuccRatio·|POP| offspring outperforming their own parents even if (MaxSelPress·|POP|) candidates have been generated, premature convergence has occurred.*

As a basic principle of this selection model a higher success ratio causes higher selection pressure. Nevertheless, higher settings of success ratio and therefore of selection pressure do not necessarily cause premature convergence as the preservation of fitter alleles is additionally supported and not only the preservation of fitter individuals.

Also it becomes possible within this model to state selection pressure in a very intuitive way that is quite similar to the notation of selection pressure in Evolution Strategies. Concretely, we define the actual selection pressure as the ratio of individuals that had to be generated in order to fulfill the success ratio to the population size.

For example, if we work with a population size of say *100* and it would be necessary to generate *2000* individuals in order to fulfill the success ratio, the actual selection pressure would have a value of *20*. Via these means we are in a position to attack several reasons for premature convergence as illustrated in the following sections. Furthermore, this strategy has proven to act as a precise mechanism for self-adaptive selection pressure steering, which is of major importance in the migration phases of parallel evolutionary algorithms. The aspects of offspring selection w.r.t. parallel GAs are combined in the parallel SASEGASA-algorithm (Affenzeller and Wagner 2003; Affenzeller and Wagner 2004a).

4. EMPIRICAL DISCUSSION

The empirical section is subdivided into three parts: The first subsection aims to highlight the main message of the paper (preservation of essential alleles) whereas the second subsection aims to touch on further subjects concerning the effects of self-adaptive offspring selection (Affenzeller 2004b). As the scope of the present work does not allow a deeper and more sophisticated analysis of different problem situations, the third part of the experimental discussion gives some references to related contributions which include a more detailed and statistically more relevant experimental discussion on the basis of several benchmark but also practical problems on which we have applied the new selection model recently.

4.1. Preservation of Essential Genetic Information

This subsection aims to point out the importance of mutation for the recovery of essential genetic information in the case of conventional GAs in order to oppose these results with the results being achieved with the enhanced selection model discussed in this paper. By reasons of compactness, the results are mainly shown on the basis of diagrams and give only a brief description of introduced operators, parameter settings, and test environments. Furthermore, the chosen benchmark instance is of rather small dimension in order to allow the observation of essential alleles during the run of the algorithm.

The results displayed in Figure 2 show the effect of mutation for reintroducing already lost genetic information. The horizontal line of the diagram shows the number of iterations and the vertical line stands for the solution quality. The bottom line indicates the global optimal solution which is known for this benchmark test case. The three curves of the diagram show the performance of a Genetic algorithm with no mutation, with a typical value of 5% mutation as well as a rather high mutation rate of 10%. For each of the three curves the lower line stands for the best solution of the actual population and the upper line shows the average fitness value of the population members. The results with no mutation are extremely weak and the quality curve stagnates very soon and far away from the global optimum. The best and average solution quality are the same and no further evolutionary process is possible – every allele is fixed and premature convergence has occurred. As already stated before, mutation is a very essential feature of standard GAs in order to avoid premature convergence. But also a rather high mutation rate of 10% produces results which are not very satisfying and indeed the best results are achieved with a mutation rate which is in a quite typical range for a lot of GA applications - namely a mutation rate of 5%.

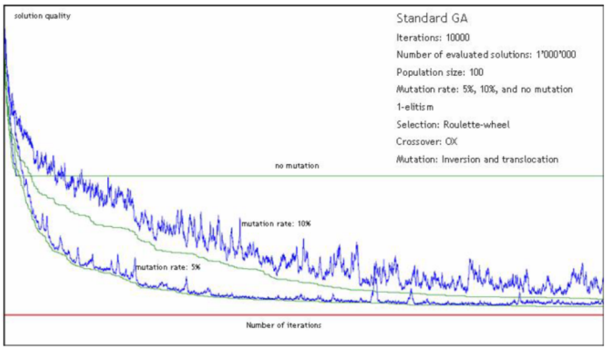


Figure 2: The Effect of Mutation in Case of a Standard GA for the ch130 benchmark TSP.

Considering a rather low dimensional standard benchmark problem like the ch130 with a unique and well known global optimal solution (a 130 city TSP taken from the TSPLib; (Reinelt 1991)) allows to consider the edges of the shortest path as the essential alleles whose preservation during the run can be observed.

The following figures indicate the spreading of essential alleles during the runs of the certain algorithms. This is visualized by inserting bar charts which have to be considered as snapshots after a certain number of iterations approximately corresponding to the position in the figure. The higher a certain bar (130 bars for a 130-city TSP) the higher the relative occurrence of the corresponding essential allele in the population.

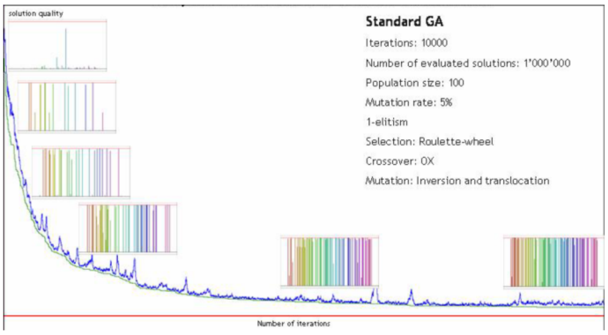


Figure 3: The Distribution of Essential Genetic Information in case of a Standard GA for the ch130 benchmark TSP

Figure 3 shows the distribution of essential alleles over the iterations for a standard GA with a mutation rate of 5%. The interesting thing is that some minor ratio of essential alleles is rapidly fixed in the population and the majority of essential alleles which are still missing have disappeared in the entire population. During the further run of the algorithm it is only mutation which can reintroduce this essential genetic information. As it could be seen in Figure 2, without mutation premature convergence would already have occurred at this early state of evolutionary search. But with an appropriate mutation rate (5% in this example) more and more essential alleles are discovered ending up with quite a good solution. But there is still a gap to the global optimum caused by those alleles which could not be recovered due to mutation. The next figures will show how the new selection concept is able to close this gap and make the algorithm much more independent of mutation.

So let us take a closer look at the distribution of essential genetic information in the population when using the enhanced selection concepts. The next curve (Figure 4) shows the quality curve and the distribution of essential alleles for 5% mutation which was able to achieve the best results in case of a standard GA.

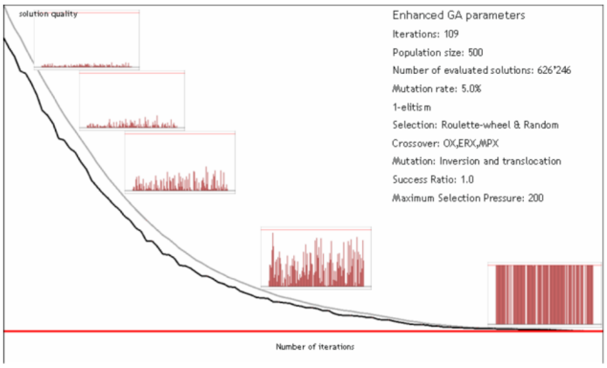


Figure 4: The Distribution of Essential Genetic Information when using the Enhanced Selection Concept considering the ch130 benchmark TSP

When applying the GA with the new offspring selection principle to the same benchmark test case one can see that the global optimal solution is detected in only about 100 iterations. Nevertheless, the computational effort is comparable to the standard GA as much more individuals have to be evaluated at each iteration step due to the higher selection pressure. Considering the distribution of essential alleles we see a totally different situation. Almost no essential alleles get lost and the ratio of essential alleles continuously increases in order to end up with a final population that contains almost all pieces of essential genetic information and therefore achieving a very good solution. This shows that the essential alleles are preserved much more effectively and indicates that the influence of mutation should be much less. But is this really the case? In order to answer this question, let us consider the same example with the same settings - just without mutation.

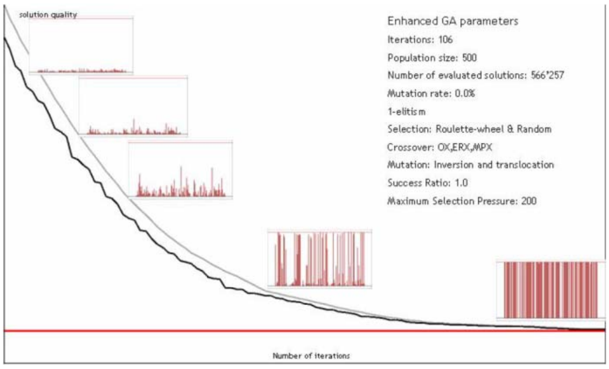


Figure 5: The Distribution of Essential Genetic Information when using Offspring Selection without Mutation

And indeed the assumption holds and also without mutation the algorithm finds a solution which is very close to the global optimum. The essential alleles interfuse the population more and more and almost all of them are members of the final population. Reconsidering the standard GA without mutation the algorithm was prematurely converging very soon with a very bad total quality.

4.2. Experimental Studies of Some More Aspects on the Basis of some TSP instances

The analysis of producible results considering various algorithms and benchmark test cases still denotes the most commonly used and possibly also the most objective way to analyze the potential of heuristic optimization techniques.

In our experiments, all computations are performed on Pentium 4 PCs with 2 GB RAM under Windows XP. The environment in which the algorithms are implemented and tested is HeuristicLab[1] (Wagner and Affenzeller 2005). For the tests shown here we have selected some rather small instances of the Travelling Salesman Problem (TSP) taken from the TSPLIB (Reinelt 1991). Reference to more sophisticated tests also on higher dimensional TSP instances are given in the next subsection. In all experiments, the results are

---

[1] More details can be found on the HeuristicLab homepage http://www.heuristiclab.com

represented as the relative difference to the best known solution defined as

$$relative\ Difference = \left(\frac{Result}{Optimal} - 1\right) \cdot$$

However, it is to be pointed out once again that the newly introduced methods are by no means restricted or somehow optimized to routing problems like the TSP. Similar effects could also be observed for other combinatorial optimization problems and especially for genetic programming applications. References are given in subsection 4.3.

In order to not dilute the effects of the different aspects too much, some selected aspects are pointed out separately in this section.

The first aspect to be considered is the effect of the enhanced selection model (offspring selection) to the quality improvement of different crossover operators. To visualize the positive effects of the new methods in a more obvious way we also present results that were generated by a classical GA with generational replacement and 1-elitism.

Remarkable in this context is the effect that also crossover operators that are considered as rather unsuitable for the TSP (Larranaga, Kuijpers, Murga and Dizdarevic 1999) achieve quite good results in combination with the new selection model. The reason for this behavior is given by the fact that in our selection principle only children that have emerged as a good combination of their parents' attributes are considered for the further evolutionary process; the success ratio levels off at a higher range. In combination with a higher upper value for the maximum selection pressure genetic search can therefore be guided advantageously also for poor crossover operators as the larger amount of handicapped offspring is simply not considered for the further evolutionary process.

Additionally to the already mentioned aspect that the enhanced selection mechanism is able to improve the performance of the certain crossover operators it is furthermore observable that the new self-adaptive selection model makes the performance of the GA almost independent of sexual selection in terms of qualitative performance.

Table 1: Parameter values used in the Test Runs of the several algorithms

Classical GA with several selection mechanisms (Tab.2, Tab. 3, Tab. 4)	
Generations	100.000
Population Size	120
Elitism Rate	1
Mutation Rate	0.05
Selection Operator	Roulette, Linear Rank, Random
Mutation Operator	Inversion & Translocation

GA with offspring selection in combination with several sexual selection mechanisms (Tab.5, Tab. 6, Tab. 7)	
Population Size	500
Elitism Rate	1
Mutation Rate	0.1
Selection Operator	Roulette, Linear Rank, Random
Mutation Operator	Inversion & Translocation
Success Ratio	0.7
Maximum Selection Pressure	250

In Tab. 2, 3, and 4 the results achieved with the conventional GA using either roulette-wheel (Tab. 2), linear-rank (Tab. 3) respectively random, i.e. no, (Tab. 4) selection are listed. On the other hand Tab. 5,6, and 7 show the results achieved with the enhanced self-adaptive selection concept using either roulette-wheel (Tab. 5), linear-rank (Tab. 6), or no (Tab. 7) sexual selection.

The fixed parameter values for all algorithms that were used in the different test runs of the present subsection are given in Tab. 1. All values presented in the following tables are the best resp. average values of twenty independent test runs executed for each test case.

Similar improvements of solution quality are also observable when comparing the GA using linear-rank selection with the enhanced GA using linear-ranking as the first selection step. So far the results underpin the crossover improvement not only for roulette-wheel selection but also for linear-ranking.

Especially notable is the comparison of Tab. 4 and Tab. 7. Firstly, it is barely remarkable that a GA with no (i.e. random) sexual selection is unable of producing high-quality results. So the results of Tab. 4 are in the region of random search which is caused by the 1-elitism (the only goal-oriented force under these settings). What is really remarkable is that the results of the new GA with enhanced selection and no sexual selection (Tab. 7) are about the same than the results obtained with roulette-wheel respectively with linear-rank as the first selection step. This observation supports the theory of population genetics that sexual selection really plays a rather inferior role in the natural selection process.

Table 2: Experimental Results achieved with the Classical GA using Roulette-Wheel Selection

Results for standard GA with proportional selection				
Problem	Crossover	Best	Av.	Eval. Sol.
berlin52	OX	0.00	3.76	12.000.00
	ERX	5.32	7.73	12.000.00
	MPX	21.74	26.52	12.000.00
ch130	OX	3.9	5.41	12.000.00
	ERX	142.57	142.62	12.000.00
	MPX	83.57	83.57	12.000.00
kroA200	OX	3.14	4.69	12.000.00
	ERX	325.92	336.19	12.000.00
	MPX	146.94	148.08	12.000.00

Table3: Experimental Results achieved with the Classical GA using Linear Rank Selection

Results for standard GA with linear rank selection				
Problem	Crossover	Best	Av.	Eval. Sol.
berlin52	OX	0.00	5.40	12.000.00
	ERX	2.52	4.58	12.000.00
	MPX	20.9	27.31	12.000.00
ch130	OX	5.60	8.88	12.000.00
	ERX	99.18	128.47	12.000.00
	MPX	85.78	97.46	12.000.00
kroA200	OX	8.58	12.24	12.000.00
	ERX	351.41	365.8	12.000.00
	MPX	144.25	150.34	12.000.00

Table 4: Experimental Results achieved with the Classical GA using Random Selection

Results for standard GA with proportional selection				
Problem	Crossover	Best	Av.	Eval. Sol.
berlin52	OX	25.07	31.85	12.000.00
	ERX	80.54	89.96	12.000.00
	MPX	52.24	78.52	12.000.00
ch130	OX	148.54	161.77	12.000.00
	ERX	397.46	406.94	12.000.00
	MPX	252.59	286.18	12.000.00
kroA200	OX	296.22	309.71	12.000.00
	ERX	667.71	692.22	12.000.00
	MPX	420.76	464.49	12.000.00

Table 5: Experimental Results achieved with Offspring Selection and Proportional Parent Selection

Results for offspring selection GA with proportional sexual selection				
Problem	Crossover	Best	Av.	Eval. Sol.
berlin52	OX	0.00	3.88	15.964.680
	ERX	0.00	3.10	16.337.700
	MPX	0.00	1.45	11.775.071
	OX, ERX, MPX	0.00	0.72	7.204.601
ch130	OX	3.88	5.40	15.602.82
	ERX	4.02	5.30	16.920.45
	MPX	1.83	3.53	13.994.68
	OX, ERX, MPX	0.00	2.71	7.702.818
kroA200	OX	2.25	5.72	10.814.980
	ERX	5.10	5.99	18.268.888
	MPX	5.21	7.65	12.296.581
	OX, ERX, MPX	0.00	2.78	6.647.256

Table 6: Experimental Results achieved with Offspring Selection and Linear Rank Parent Selection

Results for offspring selection GA with proportional sexual selection				
Problem	Crossover	Best	Av.	Eval. Sol.
berlin52	OX	2.29	4.94	7.448.762
	ERX	0.00	1.92	399.296
	MPX	0.00	3.92	8.199.592
	OX, ERX, MPX	0.00	1.59	60.920.006
ch130	OX	3.04	7.90	2.515.637
	ERX	4.36	5.36	1.245.727
	MPX	2.22	3.61	9.029.870
	OX, ERX, MPX	2.16	2.80	61.759.48
kroA200	OX	8.14	9.30	2.011.929
	ERX	6.28	8.12	4.822.588
	MPX	5.63	6.37	8.527.427
	OX, ERX, MPX	1.75	2.79	57.493.081

Table 7: Experimental results achieved with Offspring Oelection and Random Parent Selection

Results for offspring selection GA with proportional sexual selection				
Problem	Crossover	Best	Av.	Eval. Sol.
berlin52	OX	3.09	5.62	16.045.200
	ERX	0.00	1.35	16.938.904
	MPX	0.00	3.78	19.307.034
	OX, ERX, MPX	0.00	1.45	7.233.215
ch130	OX	2.24	4.59	15.281.04
	ERX	2.27	5.20	18.840.03
	MPX	3.60	4.77	23.164.73
	OX, ERX, MPX	0.00	2.05	6.797.867
kroA200	OX	3.77	6.49	13.188.469
	ERX	118.6	121.7	28.406.603
	MPX	3.13	4.04	22.728.010
	OX, ERX, MPX	0.00	2.72	6.171.308

4.3. References to Offspring Selection Applications

The basic concepts of the enhanced selection ideas as published in the present paper have already emerged a couple of years ago. As the actual focus (like also stated in the present contribution) is to study the properties of the new selection concepts systematically, the potential w.r.t. achievable advancements in global solution quality were obvious immediately. Therefore, the main aim of the first works in this area was to check the generality of the new algorithmic concepts by applying them to various theoretically as well as practically relevant problems. And indeed this worked out very well and it was possible to demonstrate similar effects and achievements in global solution quality in various areas of application under very different problem codifications with exactly that enhanced generic selection techniques as being proposed in this paper.

While the last subsections considered only relatively small TSP instances in order to illustrate some selected aspects, journal article (Affenzeller and Wagner 2004) includes a detailed and comprehensive empirical analysis also based on TSP instances of much higher dimension. Furthermore, (Affenzeller and Wagner 2003; Affenzeller and Wagner 2004; Affenzeller 2005) give a comprehensive solution analysis based on several real valued n-dimensional test functions (like the n-dimensional Rosenbrock, Rastrigin, Griewangk, Ackley, or Schwefel's sine root function). Also here it is possible to locate the global optimal solution in dimensions up to *n=2000* with exactly the same generic extensions of the selection model as being stated here - only the crossover- and mutation-operators have been replaced with standard operators for real-valued encoding.

In (Winkler, Affenzeller and Wagner 2005; Winkler, Efendic, Affenzeller, Del Re and Wagner 2005) we report promising results achieved in the context of nonlinear structure identification based on time-series data of a diesel combustion engine. Concretely the aim of this project is the development of models for the NO_x emission. Already until now it has become possible with a GP-based approach equipped with offspring selection to identify models which are superior to the models achieved with conventional GP-techniques and also superior to machine learning techniques which have also been considered in earlier stages of this project. Such results including also extensive empirical comparisons between offspring selection GP, conventional GP and other machine learning techniques like artificial neural networks are reported for dynamical mechatronical systems as well as for well known medical benchmark data sets taken from the UCI machine learning library (Winkler, Affenzeller and Wagner 2006; Winkler, Affenzeller and Wagner 2007; Winkler, Affenzeller and Wagner 2008)[2].

5. CONCLUSION

Possibly the most important feature of the newly introduced concepts is that the achievable solution quality can be improved in a non-problem specific manner so that it can be applied to all areas of application for which the theory of Genetic Algorithms and Genetic Programming provides suitable operators. Further aspects worth mentioning concern the robustness and self-adaptiveness of the population genetics inspired measures:

Basically weak operators become powerful and the selection pressure is steered self-adaptively in a way that the amount of selection pressure actually applied is that high that further progress of evolutionary search can be achieved. Nevertheless the newly developed selection techniques are not problem specific at all (cf. subsection 4.3.). Possible future research topics in that area are certainly to open new areas of application due to the increased robustness and also more theoretical topics like the analysis of various aspects of population genetics and their interaction with concrete applications of evolutionary computation. Especially for the theory of parallel genetic algorithms the interactions between genetic drift and migration should be a very fruitful field of research.

REFERENCES

Affenzeller, M., 2001. SEGA: A hybrid superstructure upwards compatible to genetic algorithms for retarding premature convergence. *International Journal of Computers, Systems and Signals (IJCSS)*; 2 (1), 18 – 32.

Affenzeller, M., 2002. A generic evolutionary computation approach based upon genetic algorithms and evolution strategies. *Journal of Systems Science*; 28 (2), 59 – 72.

Affenzeller, M., 2005, *Population genetics and evolutionary computation: Theoretical and practical aspects*. Linz: Trauner Verlag.

Affenzeller, M., Wagner, S., 2003. SASEGASA: An evolutionary algorithm for retarding premature convergence by self-adaptive selection pressure steering. *Computational Methods in Neural Modeling, Lecture Notes of Computer Science*; 2686, 438 – 445.

Affenzeller, M., Wagner, S., 2004a. SASEGASA: A new generic parallel evolutionary algorithm for achieving highest quality results. *Journal of Heuristics, Special Issue on New Advances on Parallel Meta-Heuristics for Complex Problems*; 10 (3), 239 – 263.

Affenzeller, M., Wagner, S., 2004b. Reconsidering the Selection Concept of Genetic Algorithms from a Population Genetics Inspired Point of View. *Cybernetics and Systems 2004*, 701-706.

Cavicchio, D., 1970. *Adaptive Search using Simulated Evolution*. PhD Thesis, University of Michigan.

DeJong, K., 1975. *An Analysis of the Behavior of a Class of Genetic Adaptive Systems*. PhD Thesis, University of Michigan.

Fogel, D., 1994. An introduction to simulated evolutionary optimization. *IEEE Trans. on Neural Network*; 5 (1), 3 – 14.

Goldberg, D., 1989. *Genetic Alogorithms in Search, Optimization and Machine Learning*. Addison Wesley Longman.

Larranaga, P., Kuijpers, C., Murga, R. and Dizdarevic S., 1999. Genetic algorithms for the travelling salesman problem: A review of representations and operators. *Artificial Intelligence Review*, 13 (2), 129 – 170.

Michalewicz, Z., 1996. *Genetic Algorithms + Data Structurs = Evolution Programs*. Berlin: Springer-Verlag, 3rd edition.

Rechenberg, I., 1973. *Evolutionsstrategie*. Stuttgart: Friedrich Frommann Verlag.

Reinelt, G., 1991. TSPLIB - A traveling salesman problem library. *ORSA Journal on Computing*; 3, 376 – 384.

Schoeneburg, E., Heinzmann, F. and Feddersen, S., 1994. *Genetische Algorithmen und Evolutionsstrategien*. Addison-Wesley.

Wagner, S. and Affenzeller, M., 2005. Heuristiclab: A generic and extensible optimization environment. Adaptiveand *Natural Computing Algorithms –*

---

[2] A complete list of publications is available at http://www.heuristiclab.com/publications/

Proceedings of ICANNGA 2005, 538 – 541. 21st – 23rd March, Coimbra, Portugal.

Winkler, S.; Affenzeller, M. and Wagner, S., 2005. New methods for the identification of nonlinear model structures based upon genetic programming techniques. *Journal of Systems Science*, 31, 5 – 13.

Winkler, S.; Affenzeller, M. and Wagner, S., 2006. Heuristic Modeler: A multi-purpose evolutionary machine learning algorithm and its applications in medical data analysis. *Proceedings of the International Mediterranean Modelling Multiconference I3M 2006*, 629 – 634. 4th Oct. – 6th Oct., Barcelona, Spain

Winkler, S., Affenzeller, M. and Wagner, S., 2007, Advanced genetic programming based machine learning. *Journal of Mathematical Modelling and Algorithms*, 6 (3), 455 – 480.

Winkler, S., Affenzeller, M. and Wagner, S., 2008, Offspring selection and its effects on genetic propagation in genetic programming based system identification. *Cybernetics and Systems 2008*, 2, 49-54.

Winkler, S.; Efendic,H.; Affenzeller, M.; Del Re, L. and Wagner, S., 2005, On-line modeling based on genetic programming. *Proceedings of the 1st International Workshop on Automatic Learning and Real-Time (ALaRT'05)*, 119 – 130. 7th – 8th Sept. 2005, Siegen; Germany.

AUTHORS BIOGRAPHY

MICHAEL AFFENZELLER has published several papers and journal articles dealing with theoretical aspects of evolutionary computation and genetic algorithms. In 1997 he received his MSc in mathematics and in 2001 his PhD in computer science, both from JKU Linz, Austria. He is professor at the Upper Austria University of Applied Sciences (Campus Hagenberg) and associate professor at the Institute of Formal Models and Verification at JKU Linz since his habilitation in 2004.

STEFAN WAGNER received his MSc in computer science in 2004 from Johannes Kepler University Linz, Austria. He now holds the position of an associate professor at the Upper Austrian University of Applied Sciences (Campus Hagenberg). His research interests include evolutionary computation and heuristic optimization, theory and application of genetic algorithms, machine learning and software development.

STEPHAN M. WINKLER received his MSc in computer science in 2004 and his PhD in technical sciences in 2008, both from JKU Linz, Austria. His research interests include genetic programming, nonlinear model identification and machine learning. Currently he is research associate at the Research Center Hagenberg of the Upper Austrian University of Applied Sciences, working on the research program L284-N04 “GP-Based Techniques for the Design of Virtual Sensors”, a research project funded by the Austrian Science Fund (FWF).

INCORPORATING PHYSICAL KNOWLEDGE ABOUT THE FORMATION OF NITRIC OXIDES INTO EVOLUTIONARY SYSTEM IDENTIFICATION

Stephan Winkler[a], Markus Hirsch[b], Michael Affenzeller[c], Luigi del Re[d], Stefan Wagner[e]

[a], [c], [e] Heuristic and Evolutionary Algorithms Laboratory
School of Informatics, Communications and Media; Upper Austria University of Applied Sciences
Softwarepark 11, 4232 Hagenberg, Austria

[b] Linz Center of Mechatronics
Altenbergerstraße 69, 4040 Linz, Austria

[d] Institute for Design and Control of Mechatronical Systems
Johannes Kepler University Linz
Altenbergerstraße 69, 4040 Linz, Austria

[a] stephan.winkler@heuristiclab.com, [b] markus.hirsch@lcm.at,
[c] michael.affenzeller@heuristiclab.com, [d] luigi.delre@jku.at, [e] stefan.wagner@heuristiclab.com

ABSTRACT

Genetic programming (GP) is an evolutionary optimization method that has already been used successfully for solving data mining problems in the context of several scientific domains. For example, the identification of models describing the nitric oxides (NO_x) emissions of diesel engines has been investigated intensively, very promising results were obtained using GP.

In the standard GP process, all model structures (as well as parameter settings) of models are created during an evolutionary process; populations of models are evolved using the genetic operators crossover, mutation and selection. In this paper we discuss several possibilities how a priori knowledge can be integrated into the GP process; we have used physical knowledge about the formation of NO_x emissions in a BMW diesel engine, test results are given in the empirical tests section.

Keywords: system identification, genetic programming, incorporation of physical knowledge

1. GENETIC PROGRAMMING

Genetic programming (GP) is an optimization technique based on the theory of genetic algorithms, designed for automatically creating programs that solve a given problem situation. A genetic algorithm (GA) works with a set of candidate solutions called population; during the execution of the algorithm each individual has to be evaluated, which means that a value indicating the fitness is returned by a fitness function. New individuals are created on the one hand by combining the genetic make-up of two parent solution candidates producing a new child, and on the other hand by mutating some individuals, i.e. changing randomly chosen parts of genetic information. Beside crossover and mutation, the third decisive aspect of genetic algorithms is selection: Usually, the individual's probability to propagate its genetic information to the next generation is relative to its fitness; the better a solution candidate's fitness value, the higher the probability, that its genetic information will be included in the next generation's population. This procedure of crossover, mutation and selection is repeated over many generations until some termination criterion is fulfilled.

Similar to GAs, GP works by imitating aspects of natural evolution to generate a solution that maximizes (or minimizes) some fitness function; a population of solution candidates evolves through many generations towards a solution using evolutionary operators; the main difference is that, whereas GAs are intended to find an array of characters or integers representing the solution of a given problem, the goal of a GP process is to produce a computer program solving the problem at hand. Typically the population of a GP algorithm contains a few hundreds or even thousands of individuals and evolves through the action of operators known as crossover, mutation and selection. Figure 1 visualizes how the GP cycle works (Langdon and Poli 2002): As in every evolutionary process, new individuals are created and tested, and the fitter ones in the population succeed in creating children of their own; unfit ones die and are removed from the population.

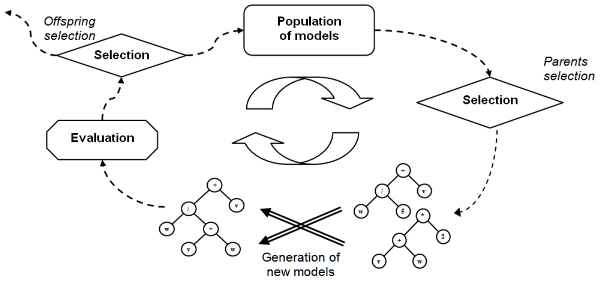


Figure 1: The extended genetic programming cycle.

2. GP BASED SYSTEM IDENTIFICATION

In general, the process of building dynamic mathematical models from measured data is called system identification (Ljung 1999). In this context, a dynamical model is a mathematical description of the dynamic behavior of a system or process which is to be identified.

The main goal here is to determine the relationship of a dependent (target) variable t to a set of specified independent (input) variables x. Thus, what we want to get is a function f that uses x and a set of coefficients w such that

$$t = f(x, w) + \varepsilon \qquad (1)$$

where ε represents the error (noise) term. Applying this procedure we assume that a model can be created with which it will also be possible to predict correct outputs for other data examples (test sample); from the training data we want to generalize to situations not known (or allowed to analyze) during the training phase.

Genetic programming has been repeatedly used successfully for building formulas that describe the behavior of systems from measured data, see for example (Koza 1992; Kejzer and Babovic 1999; Langdon and Poli 2002; Alberer, del Re, Winkler, and Langthaler 2005; del Re, Langthaler, Furtmüller, Winkler, and Affenzeller 2005; Winkler, Affenzeller, and Wagner 2006; Winkler, Affenzeller, and Wagner 2007). When it comes to evaluating a model *m* (a solution candidate in a GP based modeling algorithm, e.g.), the formula has to be evaluated on a certain set of evaluation (training) data *X* yielding the estimated values *E(m,X)*. These estimated target values are compared to the original values *T*, i.e. those which are known from data retrieval (experiments) or calculated applying the original formula to *X*. This comparison is done calculating the error between original and calculated target values; there are several ways how to measure this error, one of the simplest and probably most frequently used one being the mean squared error function.

Since 2002 we have been developing a GP based structure identification framework which also used these further developed selection principles. The HeuristicLab (Wagner and Affenzeller 2005), a framework for prototyping and analyzing optimization techniques for which both generic concepts of evolutionary algorithms and many functions to evaluate and analyze them are available, is the basis for this implementation.

3. INCLUDING A PRIORI KNOWLEDGE INTO GENETIC PROGRAMMING BASED SYSTEM IDENTIFICATION

A lot of research work has already been done regarding the use of already existing knowledge and known constraints in evolutionary system identification. Keijzer and Babovic (Keijzer and Babovic 1999), for example, describe the design of dimensionally aware GP; here, the fact that physical measurements are generally accompanied by their units of measurement is utilized leading to an extension of GP that considers the information given by the units of measurement.

In many modeling problem situations there is at least partial knowledge available about the system's structure. If the whole structure was known, then we would not necessarily need a structural system identification method as GP; but, as already insinuated, we often only know something about a certain part of the system at hand, but not the total system's structure; examples are shown in Figure 2.

Three possibilities how a priori knowledge can be incorporated into genetic programming based system identification are to be described in the following:

1. *Introduction of synthetic variables*: The most simple way to handle case 1 is to introduce an additional variable into the data base; this new variable's values are calculated according to the subsystem's model. The modeling process is thus able to incorporate this synthetic variable into models for the total system. This procedure is of course applicable and frequently used for any modeling approach. Still, subsystems as described in modeling case 2 cannot be handled using this approach since not all inputs for the modeled unit are known.
2. *Seeding parts of the population*: A genetic programming specific possibility to handle case 2 is to model the known part of the subsystem as a GP model (formula) m and to inject it into the population intentionally. This injection can be done during the population initialization phase as well as in any other phase of the GP process; in any case a certain number of individuals in the population or of the models created by crossover or mutation has to be replaced by m. For the particular example given in the right part (b) of Figure 2 this model could be for example /(+(X1, X3, 0), 1); the rest of m's inputs has to be modeled by the evolutionary process, the placeholder terminals 0 and 1 should then be replaced by appropriate subsystem representations. Furthermore, this partial model can be (by crossover) inserted into other models and so become a part of the total system's model. This model is shown in the left part (a) of Figure 2. Still, this approach comes with two major drawbacks: On the one hand, the models inserted into the population could be assigned very low fitness values and might thus be eliminated out of the population immediately. On the other hand, the models inserted into the population could be assigned very high fitness values, especially when the core of the system is modeled very accurately. The problem here is that these super-individuals could be so dominant in comparison to all other models, and this could have the effect that the population

immediately converges to a local optimum so that premature convergence could happen.

3. *Introduction of particular definitions in the functional base:* The most flexible possibility is surely to introduce particular functions and terminals with appropriate parent and child relationship definitions.

By doing so, any given subsystem can be modeled with optional references to system inputs; by using the respective functions in the genetic programming process, the so modeled a priori knowledge can be incorporated. This procedure might seem to be a bit cumbersome as it would be easier to program functions that have direct access to the data and to use the variables' values directly without needing additional terminals. Still, we have chosen to stick strictly to the original definition of functions as units processing results of other functions or terminals; this is why this approach has been implemented in this manner in HeuristicLab even though there would not have been a technical reason not to provide functions with access to the data basis.

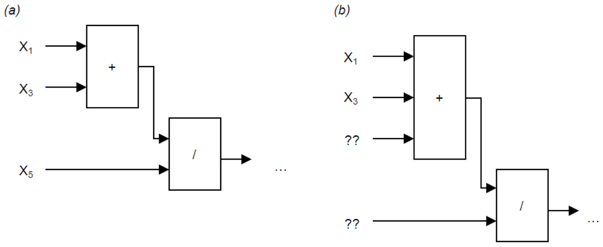


Figure 2: A priori knowledge about the structure of a system.

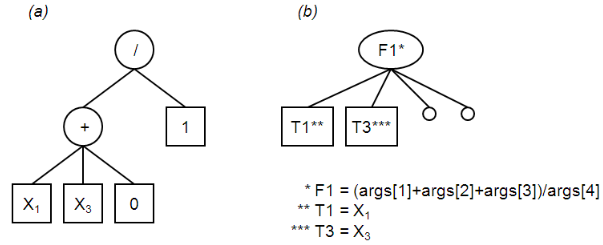


Figure 3: Models representing a priori knowledge given in Figure 2.

4. FORMATION OF NITRIC OXIDES IN DIESEL ENGINES

In fact, research regarding the formulation of dynamic models describing for NO_x emissions of diesel engines has been done since several decades, so there is already a lot of physical and chemical knowledge available for this modeling task:

As Warnatz, Maas and Dibble explain in (Warnatz, Maas, and Dibble 1996), the products of combustion are distinctly identified as a severe source of environmental damage, especially caused by increased combustion of hydrocarbon fuels. The major combustion productions, especially carbon dioxide and water, have long been considered rather "harmless"; now, carbon dioxide is more and more seen as a significant source of problems regarding the atmospheric balance and greenhouse effect. NO_x are less obvious products of combustion; within the last half of the twentieth century it has become apparent that NO and NO_2, collectively called NO_x, are major contributors to photochemical smog and ozone in the troposphere (Seinfeld 1986). Gaining knowledge regarding the production process of NO_x is therefore of great interest and researchers search for models for the production of these pollutants in order to find new ways how to minimize them (Warnatz, Maas, and Dibble 1996).

Based on physical models summarized in (Warnatz, Maas, and Dibble 1996) we have defined the following model describing the production of NO_x depending on measurable engine parameters:

$$MAF^* = \frac{MAF}{N} \cdot \frac{1000}{6} \left[\frac{kg/_h}{U/_{min}} \cdot \frac{1000}{60} = \frac{g}{u} \right] \tag{2}$$

$$NO_x = e^{(qMI \cdot (\alpha \cdot pMI + \beta \cdot \frac{1}{N} + \gamma \cdot MAF^*))} \tag{3}$$

where

- MAF is the amount of fresh air in the engine's intake manifold (MAF here stands for manifold air flow),
- N the engine speed,
- MAF* is the amount of fresh air divided by the rotational frequency and converted to the amount of air per combustion cycle,
- qMI the injected fuel mass per cycle,
- pMI is the crankshaft angle φ before the top dead centre of the piston where fuel injection starts, and
- α, β and γ are parameters which have to be identified.

Figure 4 shows a graphical representation of this semi-abstract model structure:

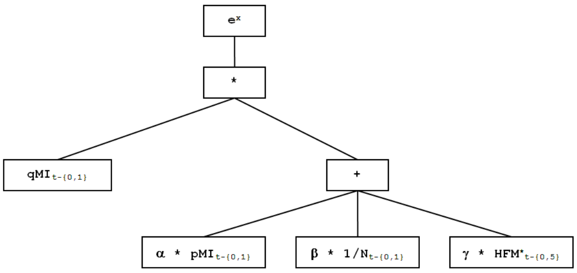


Figure 4: Model representing the physical knowledge available for the formation of NO_x emissions.

5. EMPIRICAL TESTS

The data set we have used for testing the enhanced GP strategies described in the previous sections contains the measurements taken from a 2 liter 4 cylinder BMW diesel engine at a dynamical test bench (simulated vehicle: BMW 320d Sedan). The mean engine speed was set to 2,200 revolutions per minute, and in each engine cycle 15mg fuel were injected. Several emissions (including NO_x, CO and CO_2) as well as several other engine parameters were recorded over approximately 18.3 minutes at 100 Hz and then downsampled to 10 Hz, yielding a data set containing approximately 11,000 samples. 40 signals were recorded, but only 9 variables were considered by the identification algorithm. The reason for this is that information about other emissions should not be incorporated in the model because of redundancies and relatively high costs of exhaust sensors – an emission model using other emission measurements is much easier to be found, but not very significant. Therefore we have only used parameters which are directly measured from the engine's control unit and not in any sense connected to emissions (as for example oil temperature, air pressure, injection parameters etc.).

As we now know about the physical knowledge available in context with formation of NO_x during combustion in diesel engines, there are several ways how we can make this information available for the GP process.

First and most obviously, Formula (2) describing the calculation of the auxiliary variable MAF* can be used for defining a new variable; as there are no parameters to be fixed, this new variable can be introduced into the data base immediately. The GP process is therefore able to use this information simply by using this new variable, i.e. by creating models that reference the variable MAF*.

The incorporation of the information given in Formula (3) is a bit more complicated as it includes parameters which are not known. The first possibility is to seed the population using a stub of the model already known. Of course, this brings along the problem that the unknown parameters included in the model, namely α, β and γ, have to be initialized using some arbitrary, but fixed values; we initially set those parameters to 0.1 and expect the evolutionary optimization process to tune the values so that improved model structures are evolved.

An alternative method is to create an artificial function that represents the structure of the knowledge available. So we define the following additional items that are to be added to the functional basis used by the GP process: The function definition "PKfuncNOx" represents the main part of the model. On the one hand it expects the input variables qMI, pMI, 1/N (N^{-1}) and MAF* as inputs at indices 0, 2, 4 and 6; on the other hand it also expects 5 more inputs that are processed as coefficients (at indices 1, 3, 5 and 8) or an additional term at index 7. When called with the expected inputs $I_{0 \ldots 8}$, this function returns the result of the expression defined in (4) and graphically displayed in Figure 5:

$$e^{I_0 \cdot (I_1 \cdot I_2 + I_3 \cdot I_4 + I_5 \cdot I_6 + I_7) \cdot I_8} \qquad (4)$$

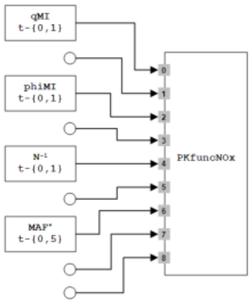


Figure 5: Terminal definitions and the "PKfuncNOx" function representing physical knowledge about the formation of NO_x emissions.

Table 1: Test strategies for incorporating physical knowledge about the formation of NO_x in the GP process.

Index	Parameters
I	No additional information.
II	Use of additional variable MAF*,
III	Use of additional variable MAF* as well as stub model
IIIa	Seed model in 20% of the initial population
IIIb	Seed model in 60% of the initial population
IIIc	Seed model in 100% of the initial population
IV	Use of additional variable MAF* as well as stub model
IVa	Seed model in 10% of the initial population and with 10% probability in main loop
IVb	Seed model in 20% of the initial population and with 20% probability in main loop
IVc	Seed model in 50% of the initial population and with 30% probability in main loop
V	Use of additional variable MAF* and "PKfuncNOx" function
Va	No manipulation of the GP-process
Vb	Introduction of "PKfuncNOx" function into solutions by special mutation operator; probability: 15%

We have applied all strategies presented in Section 3 for incorporating knowledge about the formation of NO_x. In Table 1 we summarize the test strategies actually applied; in all cases we used GP including strict offspring selection (Affenzeller, Wagner, and Winkler 2005), single-point crossover and 12% mutation.

All test strategies were executed 5 times independently using the first 5,500 samples as training data, 3000 samples as validation and the remaining samples as test data. In all test runs we collected those models that performed best on validation data and evaluated them on test data; the results of these test evaluations are summarized in Table 2 where we give statistics about the mean squared errors on test data. Figure 6 illustrates a model which was returned as best model with respect to fit on validation data in one of the test runs in series IVb. Obviously, the given model structures that were inducted into the GP processes in series IV have been used and are incorporated in the model's structure. Figure 7 shows the evaluation of this model on the whole data set.

In (Winkler 2008) the reader can find much more details about the evaluation of these test runs. For example, extensive investigations regarding population dynamics can be found in this thesis as well as further background about test bench setup and the data set used.

Table 2: Quality of results (average and standard deviation), evaluation on test data.

Strategy	Test quality	
	Mean value	Standard deviation
I	0.004359	0.00033
II	0.003901	0.00037
III		
IIIa	0.00388	0.00022
IIIb	0.00408	0.00038
IIIc	0.00429	0.00031
IV		
IVa	0.00382	0.00032
IVb	0.00348	0.00020
IVc	0.00384	0.00027
V		
Va	0.00409	0.00032
Vb	0.00388	0.00028

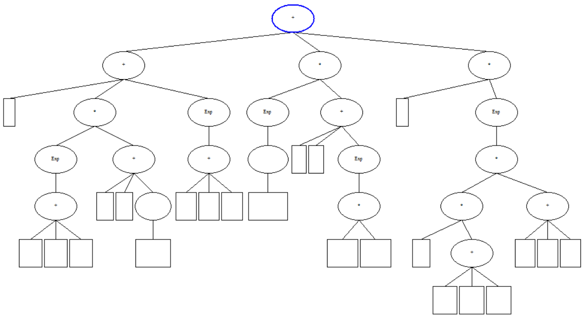


Figure 6: Best model produced for the NO_x data set: The given a priori knowledge has been incorporated as sub-trees of the returned model structure.

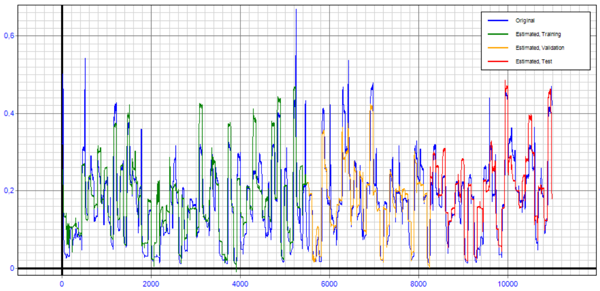


Figure 7: Evaluation of the finally retrieved model.

6. CONCLUSION

In this paper we have summarized test results for an example for the introduction of a priori about the system which is to be identified: Virtual sensors for the NO_x emissions of a BMW diesel engine have been created using physical knowledge. Three different ways how to introduce additional knowledge into the GP based learning process have been discussed and tested:

- If partial knowledge can be formulated by equations without variable parameters, then additional variables can be formed. Of course, this approach can be used in any machine learning approach; in this example application, the virtual variable MAF* has been formed and added to the problem data set, leading to increased model qualities.
- Alternatively, model structures representing partial knowledge can also be introduced into the GP process by seeding parts of the initial population or by repeatedly inserting them into the main GP loop (before crossover and mutation operations). In our example application, this was also successfully done; of course, if this forceful introduction is done too often, then population diversity can be lost leading to worse results.
- The third possibility discussed and tested here is the formation of complex functions representing partial knowledge; the genetic process is then supposed to form models that include these functions. Unfortunately, exactly this approach did not really work fine in this exemplary application: On the one hand, without manipulating the GP process, the function designed in this example died off almost completely, and on the other hand forceful introduction of this function into the existing models had negative effects on population diversity as well as on results quality.

ACKNOWLEDGMENTS

The work described here was done within the research project L284-N04 "GP-Based Techniques for the Design of Virtual Sensors" sponsored by the Austrian Science Fund (FWF). Furthermore, this work was supported by the Linz Center of Competence in Mechatronics (LCM) under grant LCM-001.

REFERENCES

Affenzeller, M., Wagner, S., and Winkler, S., 2005. Goal-oriented preservation of essential genetic information by offspring selection. *Proceedings of the Genetic and Evolutionary Computation Conference 2005*, pp. 1595–1596. June 25-29, 2005, Washington, D.C., USA.

Alberer, D., del Re, L., Winkler, S., and Langthaler, P., 2005. Virtual sensor design of particulate and nitric oxide emissions in a DI diesel engine. *Proceedings of the 7th International Conference on Engines for*

Automobile ICE 2005, paper number 2005-24-063. September 11-16, 2005, Capri, Italy.

del Re, L., Langthaler, P., Furtmüller, C., Winkler, S. and Affenzeller, M., 2005. NO_x Virtual Sensor Based on Structure Identification and Global Optimization. *Proceedings of the SAE World Congress 2005*, paper number 2005-01-0050. April 11-14, 2005, Detroit, MI, USA.

Koza, J., 1992. *Genetic Programming: On the Programming of Computers by Means of Natural Selection*. Cambridge, MA, USA: The MIT Press.

Keijzer, M. and Babovic, V., 1999. Dimensionally aware genetic programming. *Proceedings of the Genetic and Evolutionary Computation Conference 1999*, pp. 1069–1076. July 13-17, 2005, Orlando, FL, USA.

Langdon, W.B. and Poli, R., 2002. *Foundations of Genetic Programming*. Berlin, Germany: Springer Verlag.

Ljung, L., 1999. *System Identification - Theory For the User*, 2nd edition. Upper Saddle River, NJ, USA: PTR Prentice Hall.

Seinfeld, J.H., 1986. *Atmospheric Chemistry and Physics of Air Pollution*. Somerset, NJ, USA: John Wiley and Sons, Inc.

Wagner, S. and Affenzeller, M., 2005. HeuristicLab: A generic and extensible optimization environment. *Proceedings of the International Conference on Adaptive and Natural Computing Algorithms*, pp. 538-541. March 21-23, 2005, Coimbra, Portugal.

Wagner, S. and Affenzeller, M., 2005. SexualGA: Gender-Specific Selection for Genetic Algorithms. *Proceedings of the 9th World Multi-Conference on Systemics, Cybernetics and Informatics 2005*, pp. 76-81. July 10-13, 2005, Orlando, FL, USA.

Warnatz, J., Maas, U. and Dibble, R.W., 1996. *Combustion - Physical and Chemical Fundamentals, Modeling and Simulation, Experiments, Pollutant Formation*. Berlin, Germany: Springer Verlag.

Winkler, S., Affenzeller, M. and Wagner, S., 2006. HeuristicModeler: A Multi-Purpose Evolutionary Machine Learning Algorithm and its Applications in Medical Data Analysis. *Proceedings of the International Mediterranean Modelling Multiconference I3M 2006*, pp. 629-634. October 4-6, 2006, Barcelona, Spain.

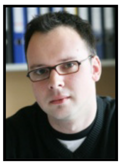
Winkler, S., Affenzeller, M. and Wagner, S., 2007. Advanced Genetic Programming Based Machine Learning. *Journal of Mathematical Modelling and Algorithms*, 6: 455-480.

Winkler, S., Affenzeller, M. and Wagner, S., 2007. Selection Pressure Driven Sliding Window Genetic Programming. *Computer Aided Systems Theory - EUROCAST 2007, Lecture Notes in Computer Science* 4739: 788-795.

Winkler, S., 2008. *Evolutionary System Identification - Modern Concepts and Practical Applications*. PhD Thesis, Johannes Kepler University Linz. April 2008, Linz, Austria.

AUTHORS BIOGRAPHIES

STEPHAN M. WINKLER received his MSc in computer science in 2004 and his PhD in engineering sciences in 2008, both from JKU Linz, Austria. His research interests include genetic programming, nonlinear model identification and machine learning. Currently he is research associate at the Research Center Hagenberg of the Upper Austrian University of Applied Sciences, working on the research program L284-N04 “GP-Based Techniques for the Design of Virtual Sensors”, a research project funded by the Austrian Science Fund (FWF).



MARKUS HIRSCH received his MSc in Mechatronics from JKU Linz, Austria in 2006; since then he has been research associate at the Linz Center of Competence in Mechatronics (LCM), also within the research program L284-N04 “GP-Based Techniques for the Design of Virtual Sensors”. His research interests include system identification, nonlinear modeling, design of experiment, biological systems, automotive control, and engine control and optimization.

MICHAEL AFFENZELLER has published several papers and journal articles dealing with theoretical aspects of evolutionary computation and genetic algorithms. In 1997 he received his MSc in mathematics and in 2001 his PhD in engineering sciences, both from JKU Linz, Austria. He is professor at the Upper Austria University of Applied Sciences (Campus Hagenberg) and associate professor at the Institute of Formal Models and Verification at JKU Linz since his habilitation in 2004.

LUIGI DEL RE is professor at Johannes Kepler University Linz and head of the Institute for Design and Control of Mechatronical Systems. He received his MSc in electronic engineering in 1981 and his PhD degree in technical sciences in 1990, both from ETH Zürich, Switzerland. His research interests include nonlinear control, parameter and structure identification, automotive control, model predictive control, and systematic design assessment.

STEFAN WAGNER also received his MSc in computer science in 2004 from Johannes Kepler University Linz, Austria. He currently holds the position of an associate professor at the Upper Austrian University of Applied Sciences (Campus Hagenberg). His research interests include evolutionary computation and heuristic optimization, theory and application of genetic algorithms, machine learning and software development.

SIMULATION OPTIMIZATION WITH HEURISTICLAB

Andreas Beham[(a)], Michael Affenzeller[(b)], Stefan Wagner[(c)], Gabriel K. Kronberger[(d)]

[(a)(b)(c)(d)]Upper Austria University of Applied Sciences, Campus Hagenberg
School of Informatics, Communication and Media
Heuristic and Evolutionary Algorithms Laboratory
Softwarepark 11, A-4232 Hagenberg, Austria

[(a)]andreas.beham@heuristiclab.com, [(b)]michael.affenzeller@heuristiclab.com
[(c)]stefan.wagner@heuristiclab.com, [(d)]gabriel.kronberger@heuristiclab.com

ABSTRACT

Simulation optimization today is an important branch in the field of heuristic optimization problems. Several simulators include built-in optimization and several companies have emerged that offer optimization strategies for different simulators. Often the optimization strategy is a secret and only sparse information is known about its inner workings. In this paper we want to demonstrate how the general and open optimization environment HeuristicLab in its latest version can be used to optimize simulation models.

Keywords: simulation-based optimization, evolutionary algorithms, metaoptimization

1. INTRODUCTION

In Ólafsson and Kim (2002) simulation optimization is defined as "the process of finding the best values of some decision variables for a system where the performance is evaluated based on the output of a simulation model of this system". From the point of view of optimization a parameter vector is to be optimized where the components may stem from different domains such as integers, real or binary values or even items from an enumeration, strings or any array in general. Additionally there is usually a feasible region that restricts the possibilities of the parameter vector. To find the optimum to those models, there are exact algorithms, heuristics and especially metaheuristics in the toolbox of an optimization engineer: Genetic Algorithms, Tabu Search, Ant Colony Optimization, Simulated Annealing, Particle Swarm Optimization, Variable Neighborhood Search and Scatter Search to name just a few.

HeuristicLab (Wagner and Affenzeller 2005, Wagner et al. 2007) is a framework that allows users to build upon these optimization strategies or use predefined strategies that are adopted from published works. It was designed such that parameterization as well as customization of any strategy can be done via a graphical user interface (GUI). So the user does not work on source code files, but creates and modifies what is called a "workbench" file. This contains the structure and the parameters of the designed optimization strategy and can be executed in the HeuristicLab optimization environment. After the execution has terminated or was aborted the results or respectively intermediate results are saved along the workbench. So in addition to the structure and parameters, the results can also be saved in a single document, reopened at any time and examined.

2. DESCRIPTION OF THE FRAMEWORK

The user interface to an evolution strategy (ES) is shown in Figure 1. The methods defined by this interface are common to many optimization strategies and reusing them is one idea of the optimization environment.

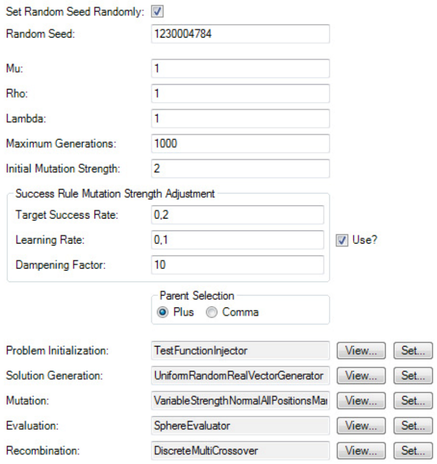

Set Random Seed Randomly: ☑
Random Seed: 1230004784
Mu: 1
Rho: 1
Lambda: 1
Maximum Generations: 1000
Initial Mutation Strength: 2
Success Rule Mutation Strength Adjustment
Target Success Rate: 0,2
Learning Rate: 0,1
Dampening Factor: 10
☑ Use?
Parent Selection
● Plus ○ Comma
Problem Initialization: TestFunctionInjector
Solution Generation: UniformRandomRealVectorGenerator
Mutation: VariableStrengthNormalAllPositionsMa
Evaluation: SphereEvaluator
Recombination: DiscreteMultiCrossover

Figure 1: Interface to configure an evolution strategy

To apply simulation-based optimization HeuristicLab needs to know about the parameter vector and any constraints that are imposed on the parameters. This is called injecting the problem. An operator provides a user interface for this task with which the parameter vector is defined and a number of constraints which are specified directly on the parameters or on the vector itself when e.g. comparisons between two parameters need to satisfy a certain criterion.

Additionally for every parameter an initialization as well as manipulation operator is defined which will perform the respective tasks during the optimization. It is also possible to add a default operator to a certain parameter which will not perform any manipulation and it is even possible to manually set a certain parameter to some value. This makes it easy to setup a number of variables for optimization, but optimize only a subset of them leaving other variables fixed. A simulation expert then is able to try to optimize only certain aspects of the simulation model while retaining the possibility to quickly include more parameters. This flexibility is likely welcomed by those who already have some ideas about the optimum of a model which they want to explore more thoroughly. It is also possible to let the optimizer use all parameters and get a general idea of achievable quality.

After the parameter vector is defined it is necessary to specify the communication between optimizer and simulator. Azadivar (1999) already mentions that interfacing simulation and optimization is not always an easy task. The generic concepts that HeuristicLab provides for designing optimization strategies also influenced the decision to build on a generic and customizable interface to communicate with external applications. The backbone to this interface is the design of the communication protocol. This protocol is described as a state-machine, a powerful tool to model algorithmic behavior and the communication between two peers. In the protocol editor users define states, which objects should be communicated in which states, state transitions and their condition. Finally the protocol is woven into the optimization strategy in the form of operators which can be put into any place during the execution of the optimization algorithm. See Figure 2 for an example interface to create the protocol.

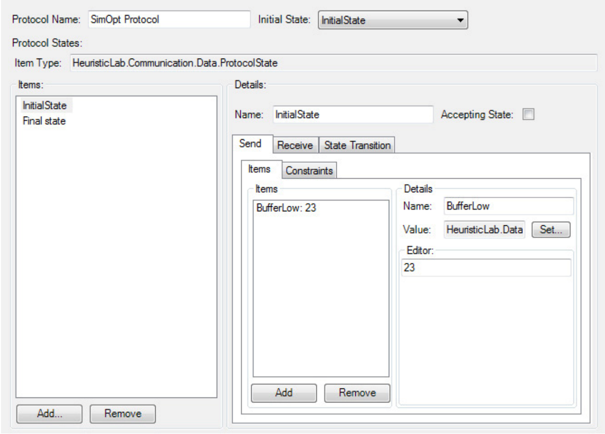

Protocol Name: SimOpt Protocol
Initial State: InitialState
Item Type: HeuristicLab.Communication.Data.ProtocolState
Items: InitialState, Final state
Name: InitialState
Accepting State: ☐
BufferLow: 23
Name: BufferLow
Value: HeuristicLab.Data
Editor: 23

Figure 2: Communication protocol interface

On the simulation side the protocol needs to be implemented as well, but because the language of choice for many simulators is C/C++ or Java and HeuristicLab builds on C# and Microsoft .NET an automated way of including the protocol into the simulation is not yet available. It is however possible to implement libraries for different programming languages that will take care of the protocol handling. Basically it is a good idea to start designing the simulation, then think about the communication with the optimizer and model the protocol using the editor in HeuristicLab.

The generality of this protocol specification does not yet forestall the actual communication topology that will be used in the communication process. As is mentioned above, transmission is implemented in operators which can be interwoven with the optimization strategy and generally part into two levels: Serialization and transmission. The more complex, but more general serialization is an XML based serialization of the objects. This is useful when the partner that deserializes the information does know about XML and can reconstruct the HeuristicLab datatypes. Another possibility is to use a simpler serialization which just outputs the variable types, names and values each on a new line and is quick to implement in different programming languages. It is also possible to add any other type of serialization by implementing new operators as plugins. Once the data has been serialized it can be transmitted and again several operators will take care of that. There are operators for StdIn/StdOut communication and operators for sending the data over a TCP network. Again, extension to any underlying network transport protocol is possible.

During the execution of the algorithm communication will be performed by a specific object which derives from the interface *IDataStream*. This interface has five methods: *Initialize*, *Connect*, *Close*, *Write* and *Read*. This object is placed within the scope tree and available to a number of operators which then call these methods.

After the communication with the simulator has been fixed the next step is to choose an optimization strategy, appropriate parameters and start optimizing.

3. EVOLUTION STRATEGY

The first ES (Rechenberg 1973) described used a population of one parent and one offspring typically denoted as (1+1)-ES. Starting from a randomly initialized parent, mutation adds Gaussian distributed noise to each value of the parameter vector. The mutated parameter-vector is evaluated and replaces the old parent if it is better than the parent; if not, the new solution is discarded. Further investigations and developments resulted in the $(\mu+\lambda)$-ES (Schwefel 1987) where λ new individuals are generated for the next replacement phase. This $(\mu+\lambda)$-ES is effectively an elitist algorithm and an alternative scheme called ‘comma-strategy’ is also defined. In this the μ best individuals are used only from the offspring not considering the old parents. This is denoted by (μ,λ)-ES. In this scheme λ needs to be significantly greater than μ or otherwise the search performs a random walk instead of an optimization.

The random noise added to each element of the parameter vector is normally distributed with parameters (0, σ). Rechenberg observed for the (1+1)-ES that adapting σ in the course of the evolution leads to better results; based on this observation he formulated the “1/5 success rule” that adapts σ based on the average success of the last n mutations with the goal of generating on average 1/5 successful mutations. When the ratio of successful mutations is greater than 1/5, the standard deviation should be increased while it should be decreased when the ratio of successful mutations is less than 1/5.

The evolution strategy is designed with respect to continuous parameter adjustment. To apply it on simulation-based optimization problems, however does not always rely on continuous parameters alone (Affenzeller et al. 2007). Some models and also the one we used for testing in this paper make use of discrete parameters, e.g. integers as well. The mutation operators need to adapt to these mixed-integer optimization problems. Newer mutation strategies are proposed that take into account a mixture of discrete and continuous parameters. Even more data representations, such as unordered discrete parameters, arrays, boolean values and others may be encountered in simulation-based optimization which requires the definition of further extensions to the evolution strategy. Common to all these manipulation concepts are several properties (Bäck and Schütz 1995):

- Smaller changes should be more likely than larger ones
- Change in any direction of the domain should be equally likely

4. RESULTS

4.1. Optimization of a Supply Chain Simulation Model

Results have been computed with one of the simulation models that ships with AnyLogic 6, namely the supply chain simulation model. This model consists of three buffers at each location in a supply chain: Retailer, Wholesaler and Factory. The retailer has to hold items in stock to sell to customers and orders new items from the wholesaler which in turn orders from the factory which produces items from an infinite source of instantly available raw materials. Each buffer is accompanied with two decision variables representing an upper and lower limit of the buffer. New items are ordered/produced when the stock falls below the lower limit in such quantity that it would fill the buffer to the upper limit. Orders at the factory and the wholesaler are shipped at the start of each day only. The model has constrained input parameters: The decision variables have to lie in the interval [1;200] and the lower limit must not be greater than the upper limit. The goal of the model is to reduce the total costs in the supply chain by minimizing buffer sizes in each location. Naturally, the optimum to this goal would be not to have any buffers and produce only on demand. This would however have customers wait for days to get their items and so reducing the mean customer waiting time (E[waitingTime]) is a second goal which stands in opposition to the first. A two goal approach however would require the application of multiobjective optimization techniques, so in this model the second goal is considered as constraint. The solution is described as feasible only when the mean customer waiting time is ≤ 0.001. In our example we allowed infeasible solutions during the search, but added a penalty to their quality value. For a feasible child with E[waitingTime] ≤ 0.001 the fitness function is E[DailyCosts], in case of an infeasible child, this value becomes: E[DailyCosts] * (100 + E[waitingTime]).

The optimum is to set the decision variables such that the buffer size becomes as small as possible while maintaining a low waiting time. Given these characteristics it is clear to assume that the “global best feasible” solution borders the infeasible region in the search space. It is also valid to assume that the feasible region covers a coherent space of reasonable size. It seems unlikely to find an optimal buffer configuration deep within the territory of infeasibility, e.g. with low costs (small buffers), but also low waiting time.

The size of the search space is reasonably large such that exact calculation or enumeration is not viable anymore. For each buffer there are about 20,000 possible combinations of lower and upper bounds, which amount to about 8 trillion possible combinations when considering all three buffers.

Another challenge in this model is its stochastic behavior. Customer demand is modeled by a random variable, so that a single simulation run usually is not enough to test for the feasibility of a solution or give a reliable estimate for its quality. The implications to the fitness landscape are that the region surrounding the global optimal solution(s) is highly disturbed by noise. There is even the question remaining if there can be a global optimal solution at all without a given confidence level for the feasibility of a solution. Because arrival is random even the safest solution can seem infeasible if the retailer is stormed on an unlucky day. The chance for such an event is extremely low though and thus we can only find feasible optimal solutions with regard to a certain probability.

We have performed two tests with HeuristicLab to optimize this model: In the first test we treated the simulation as deterministic model and initialized it with the same random seed in each replication, while in the second test we used a different random seed and optimized a stochastic model.

The results are compared to the commercial optimizer OptQuest which is already implemented within this sample simulation model. To each optimizer about the same amount of simulation replications is given as termination criterion. OptQuest applies a mixture of Scatter Search and Tabu Search and combines it with Artifical Neural Networks to learn about the response curve during the search (Glover et al. 1999).

4.1.1. Deterministic Simulation

Here we used a (5+10)-ES with self adaptive mutation strength adjustment as described in (Igel et al. 2006). The initial population is generated by a uniform random initialization of the decision variables within their bounds, manipulation adds a normal distributed variable N(0,1) multiplied with the current mutation strength and rounds the result to the next integer. This is not optimal as (Bäck and Schütz 1995) notes, however we are using a simpler self adaptive technique with just one adaptive parameter for the whole vector and no recombination. We found that as simple as this approach is, the results we could achieve are of good quality. The parameters for the 1/5-success rule self adaptation were set as follows: Initial mutation strength: 10, learning rate: 0.1 and damping factor: 50. It was run for a maximum of 1000 generations which amounts to about 10,000 evaluations. Because the model is deterministic no replications are made.

The best found solution in 5 runs with different random seeds has a quality value of 505.595, the parameter vector contained 43-55, 44-51 and 36-77 as the lower and upper bounds for the retailer, wholesaler and factory respectively. This has been the lowest feasible quality that we encountered during the test and it is obvious that it is likely to be infeasible in a majority of the situations. Indeed, evaluation of this setting with 100 independent replications showed that none of them achieved a feasible result. This is likely a solution which is feasible in very few cases. Other solutions obtained from these tests had slightly worse quality, but were only slightly more likely to be feasible when tested on the stochastic model. Thus the optimizer, in its best run, moved past a global optimal solution and into the region where a solution is likely to be infeasible, except that for the one random seed that the model has been initialized with it was still feasible. This shows that the optimizer will find highly specific solutions which work only under the fixed random setting. To obtain a real result of the performance, we will need to optimize the stochastic model.

4.1.2. Stochastic Simulation

As interesting and well performing as the results from the deterministic case are, the stochastic case is much closer to the real world. Its optimization however raises a new challenge. If we would evaluate a given configuration only a single time we may not have enough confidence about the feasibility of the solution to accept it into the next generation. So we need to have some confidence which configurations and their qualities to accept.

It was tried first with just 5 replications per individual and the final solutions were found to have moved into a region where solutions have a higher probability of being infeasible. Replications were then raised to 10 per evaluation and the results showed more confidence. Naturally, the more replications are made the more computational effort is necessary and we were still not satisfied fully with the confidence given this amount of replications, but then thought of a different approach instead of raising it yet another time. A new child is evaluated and judged by 10 replications, but as it becomes parent, producing offsprings over possibly many generations and thus influencing the search trajectory, it is evaluated again once in each generation and the average is computed anew. This requires us to keep track of the previous qualities and shows once again the strength of HeuristicLab where this could be prototyped easily within the running application. For the search it means that the longer a parent survives, the more evaluations it will collect and the more confidence we could gain in its quality.

To compute the results we used the same (5+10)-ES as in the deterministic case, with 10 replications per evaluation and adding another replication for each parent in each generation. The stop criterion was 50 generations at which about 5300 simulation runs have been counted. One test was set to run to 100 generations producing 10,550 simulation runs. The parameters for the 1/5-success rule self adaptation were set as follows: Initial mutation strength: 10, learning rate: 0.2 and damping factor: 10 to allow for quicker adaption given the small number of generations. OptQuest was also applied with the stopping criterion of 5300, 10,550 and one test with a maximum of 100,000 runs to see what is possible. Table 1 shows a summary of the results.

Table 1: Comparison of HeuristicLab (HL) with OptQuest (OQ) on the supply chain simulation model in AnyLogic 6

	Evaluations	Average	Best	Worst
HL	5300	550.03	540.17	564.51
	10550	537.39		
OQ	5300	578.63	571.34	585.04
	10550	567.01	559.73	580.21
	100000	548.88		

The best found solution with the (5+10)-ES in HeuristicLab was finally given more replications later and has a confidence of more than 90% in 100 additional replications. The solution had 14 replications counted by the algorithm. The parameters found were 57-60, 57-65 and 45-92 for the bounds of the retailer, wholesaler and factory respectively. The oldest solution in this run's parent population came to 28 replications, which means it survived 18 generations. Its quality was 538.81 and not much worse than the best. Its confidence was also above 80%. The best solution found by HeuristicLab with 5300 replications had a quality of 540.17 stored 17 replications during the algorithm run. It has a confidence of about 87%. The parameters are 66-66, 63-66 and 34-78. One problem that we saw was that feasibility is hard to evaluate. The optimization strategy used is very keen to walk past an optimal solution and into the region of infeasibility. Elitism may partly be blamed for this kind of behavior: Once the "right" 10 replications have been found for a good configuration it is accepted into the parent generation. The approach using reevaluations could lessen this effect somewhat. An alternative strategy would be to use less replications in the beginning, and increase this number as the search progresses.

The best result that OptQuest found had following parameters: 73-82, 60-60 and 41-74. In 100 additional tries it was found to be infeasible 10% of the time. The number of fixed replications per solution was also set to 10 similar to the ES in HeuristicLab.

Elapsed runtime is another important property of an optimization strategy and one where HeuristicLab is also competitive as optimization environment for simulation-based optimization. The workbench with 10550 replications took approximately 4 minutes to evaluate on an Intel Pentium 4 running at 3.0 Ghz. This comes to about 44 replications per second. On the same computer and using the same number of replications OptQuest, including user interface updates, finished in 3 minutes which is about 60 replications, though it is likely to be even faster when the GUI is turned off.

Nevertheless, to scale to more computational intensive simulations as well as to parallel hardware HeuristicLab also features several parallelization strategies such as threading and distributed computing and can thus talk to multiple simulation models running at once.

The results indicate that our evolution strategy performed quite successful and that it seemed adequate from the point of view of optimization complexity. One advantage with a general and open optimization environment such as HeuristicLab is that basically any optimization strategy can be tried, from the simplest greedy search to the most complex metaheuristic techniques.

4.2. Metaoptimization

Another possibility for simulation-based optimization is in metaoptimization. The goal is similar to that of simulation-based optimization, except that the simulation model is a given optimization strategy with several parameters that is applied on a given problem. A metaoptimization algorithm continuously feeds new parameters into another optimization strategy and receives a measure of its performance. The goal could be to tune the parameters so that the underlying optimization strategy finds better solutions and/or becomes more robust so that it finds them with higher probability. In our example we tried to optimize the self-adaptive parameters of an evolution strategy that is applied on a real valued optimization problem.

The learning parameters to be optimized are: The initial mutation strength, the learning rate and the damping factor. They control the mutation strength adjustment in a (1+1)-ES which is applied on a 50 dimensional sphere function. The ES is set to run for 2000 generations. The metaoptimizer in this case is a Genetic Algorithm with a population of 100, 5% mutation, 1-Elitism and Roulette-wheel selection. The crossover is a simple one point crossover that takes one part of the parameter vector from one parent and the other part from another parent.

The results in the five test runs performed indicated that several different settings have been found to achieve a good result within the 2000 generations. The best result had a very high learning rate of 0.97, an initial mutation strength of 4.70 and a damping factor of 4.28. Its quality averaged over 10 independent runs ranged from 0.0118 to 0.05800. Another metaoptimization run resulted in a best quality of 0.0139 with a learning rate of 0.58, an initial mutation strength of 6.62 and a damping factor of 4.04. Another one found a good solution with a quality ranging from 0.0147 to 0.0674 and a low learning rate of 0.21, initial mutation strength of 13.63 and damping factor of 6.37. It is difficult to draw conclusions about the direction of a best setting. One common thing that we observed while examining the found solutions is that the higher quality solutions overall had a lower damping factor than the worse quality solutions in the final population, whereas we found a damping factor too low, e.g. close to 1 also among the worse solutions in the final population. Such correlations however are not based on statistical evaluation. The amount of data generated during such metaoptimization tests could be used to gather a better understanding of the influence of the parameters on the metaheuristics applied, especially on more complex problems and strategies which do not lend well to analytical study. Metaoptimization could be used to explore the parameter space and statistical as well as machine learning methods could be used to find correlations, clusters or patterns in the data which could lead to more efficient parameter settings.

5. CONCLUSION AND FUTURE PERSPECTIVES

In this work we shared how the HeuristicLab framework can be used for simulation-based optimization and presented some results based on a basic simulation model as well as metaoptimization. The results show that metaheuristics built with this environment can be used effectively for these kind of tasks.

We also showed several interesting topics of research: Dealing with noisy optimization when optimizing stochastic simulation models on the one hand and the adaptation and application of well known heuristics for the purpose of simulation-based optimization on the other hand. Also interesting will be the topic of metaoptimization and how to combine it with simulation-based optimization as well as the topic of parallelization which goes beyond the scope of this paper.

We plan to continue and extend the work described and further test optimization strategies for the purpose of simulation-based optimization as well as seek applications of our work to simulation models from the industry to evaluate the performance of a number of possible approaches.

REFERENCES

Affenzeller, M., Kronberger, G., Winkler, S., Ionescu, M., Wagner, S. 2007. Heuristic Optimization Methods for the Tuning of Input Parameters of Simulation Models. *Proceedings of I3M 2007*, pp. 278-283. October 4-6. Bergeggi (Italy).

Azadivar, F. 1999. Simulation Optimization Methodologies. *Proceedings of the 1999 Winter Simulation Conference*, pp. 93-100. December 5-8. Phoenix (Arizona, USA).

Bäck, T., Schütz, M. 1995. Evolution Strategies for Mixed-Integer Optimization of Optical Multilayer Systems. *Evolutionary Programming IV: Proceedings of the 4th Annual Conference on Evolutionary Programming*, pp. 33-51. March 1-3. San Diego (California, USA). MIT Press, Cambridge, MA (USA).

Glover, F., Kelly, J.P., Laguna, M. 1999. New Advances for Wedding Optimization and Simulation, *Proceedings of the 1999 Winter Simulation Conference*, pp. 255-260. December 5-8. Phoenix (Arizona, USA).

Igel, C., Suttorp, T., Hansen, N. 2006. A Computational Efficient Covariance Matrix Update and a (1+1)CMA for Evolution Strategies, *Proceedings of the 8th Annual Conference on Genetic and Evolutionary Computation (GECCO'06)*, pp. 453-460. July 8-12. Seattle (Washington, USA).

Ólafsson, S., Kim, J. 2002. Simulation Optimization. *Proceedings of Winter Simulation Conference*, pp. 79-84. December 8-11, San Diego (California, USA).

Rechenberg, I. 1973. *Evolutionsstrategie*. Friedrich Frommann.

Schwefel, H-P. 1987. Collective Phenomena in Evolutionary Systems. *Preprints of the 31st Annual Meeting of the International Society for General System Research*, pp. 1025-1033. June 1-5. Budapest (Hungary).

Wagner, S., Affenzeller, M. 2005. HeuristicLab: A Generic and Extensible Optimization Environment. In Ribeiro, B. Albrecht, R.F., Dobnikar, A., Pearson, D.W., Steele, N.C., eds. *Adaptive and Natural Computing Algorithms*. Springer-Verlag New York, Inc., pp. 538-541.

Wagner, S., Winkler, S., Braune, R., Kronberger, G., Beham, A., Affenzeller, M. 2007. Benefits of Plugin-Based Heuristic Optimization Software Systems. *Lecture Notes in Computer Science 4739*, pp. 747-754. Springer-Verlag.

AUTHORS BIOGRAPHY

ANDREAS BEHAM received his MSc in computer science in 2007 from Johannes Kepler University (JKU) Linz, Austria. His research interests include heuristic optimization methods and simulation-based as well as combinatorial optimization. Currently he is a research associate at the Research Center Hagenberg of the Upper Austria University of Applied Sciences (Campus Hagenberg).

MICHAEL AFFENZELLER has published several papers and journal articles dealing with theoretical aspects of genetic algorithms and evolutionary computation in general. In 1997 he received his MSc in Industrial Mathematics and in 2001 his PhD in Computer Science, both from Johannes Kepler University Linz, Austria. He is professor at the Upper Austria University of Applied Sciences (Campus Hagenberg) and associate professor at the Institute of Formal Models and Verification at the Johannes Kepler University Linz, Austria since his habilitation in 2004.

STEFAN WAGNER received his MSc in computer science in 2004 from Johannes Kepler University Linz, Austria. He currently holds the position of an associate professor at the Upper Austria University of Applied Sciences (Campus Hagenberg). His research interests include evolutionary computation, heuristic optimization, theory and application of genetic algorithms, machine learning and software development.

GABRIEL K. KRONBERGER received his MSc. in computer science in 2005 from Johannes Kepler University Linz, Austria. His research interests include parallel evolutionary algorithms, genetic programming, machine learning and data-mining. Currently he is a research associate at the Research Center Hagenberg of the Upper Austrian University of Applied Sciences (Campus Hagenberg).

The Web-pages of the authors as well as further information about HeuristicLab and related scientific work can be found at `http://www.heuristiclab.com`.

RECOGNITION OF TRANSITIONS BETWEEN DIFFERENT PHASES OF THE PRODUCT LIFE CYCLE

Anatoly Sukov

Riga Technical University, Department of Modelling and Simulation, 1 Kalku Street, Riga LV-1658, Latvia

Anatolijs.Sukovs@cs.rtu.lv

ABSTRACT

Management of the product life cycle and of the corresponding supply network largely depends on information in which specific phase of the life cycle one or another product is. Finding a phase of the product life cycle can be interpreted as recognition of transitions between phases of life of these products. This paper provides a formulation of the above mentioned task of recognition of transitions and presents the structured data mining system for solving that task. The developed system is based on the analysis of demand of historical products and on information about transitions between phases in those products. The paper describes necessary data pre-processing and transformation steps, whose aim is to create a possibility of discovering rules in those data. The created rule discovering framework does not need a complicated realization, because the rules themselves can be discovered by a well-known and available classifier.

Keywords: data pre-processing, product life cycle, rule induction, classification

1. INTRODUCTION

Any created product has a certain life cycle. The term “life cycle” is used to describe a period of product life from its introduction on the market to its withdrawal from the market. Life cycle can be described by different phases: traditional division assumes such phases like introduction, growth, maturity and decline (Kotler and Armstrong 2006).

For products with conditionally long life cycle, it is possible to make some simplification, and, from the viewpoint of the dynamics of demand changes, the above mentioned phases can be merged into three. The first phase corresponds to introduction and growth; it is gradual or headlong growing of demand value in each of subsequent periods of time. The second and the third phases are the same - maturity and decline (also known as end-of-life). From the viewpoint of the management it is important to know, in which particular phase the product is. One of applications of that knowledge is selection of the production planning policy for the particular phase (Merkuryev, Merkuryeva, Desmet, and Jacquet-Lagrèze 2007). For example, for the maturity phase in case of determined demand changing boundaries it is possible to apply cyclic planning (Campbell and Mabert 1991), whereas for the introduction and decline phase an individual planning is usually employed.

From the side of data mining (Han and Kamber 2006) information about demand of particular product is time series, in which demand value is, as a rule, represented by the month. If there are different phases of the product life cycle, then there are different periods, in which transitions between these phases occur. Correspondingly, the task of recognition of the current phase for particular product consists of recognition of transitions between different phases of the product life cycle. It should be noted that such classical methods like Boston matrix (Kotler and Armstrong 2006) are unlikely worth to use for analysis of separate products, because those methods simplify the situation too much and use generalized information about groups of products. In such way, the need for creating a stable model for recognition of transitions between phases for each separate product exists.

This paper proposes a data mining framework for creating a model for recognition of the above described transitions. The model is based on the available in an enterprise database about the demand on historical products. The term “historical products” is used to describe those products, which already have transition of interest, for example, from introduction to maturity phase. After certain pre-processing steps and specific transformation of historical data, it is possible to apply a classifier, which is based on rule induction. As a result, the discovered rules are used for recognition of transitions between different phases of the analyzed product.

Successive parts of the work are organized as follows. Section 2 provides a more detailed definition of the transitions recognition task, and indicates necessary conditions and data for solving that task. Section 3 describes necessary pre-processing steps and presents specification of transformation of historical data. Selection of the classifier, and also discussion on the parameter setting, which influence the result of classifier work, are shown in Section 4. The results of real data experiments are discussed in Section 5. Section 6 summarizes features of the proposed framework and outlines possible directions of future research.

2. PROBLEM STATEMENT

If we assume that there are three different phases in product life cycle, namely, introduction, maturity and end-of-life, then two transitions are possible. The first transition is from introduction phase to maturity phase, and the second – from maturity to the product's end-of-life. Assume there are two products, for one of which information about the point of that product transition from introduction phase to maturity phase is available. The second product is just starting its life on the market and is similar to the first product with regard to the relative demand (see Fig. 1). If we are guided, for example, by the similarity degree of those two products, we will most likely make a decision that for the second product the transition will occur at the same time as for the other.

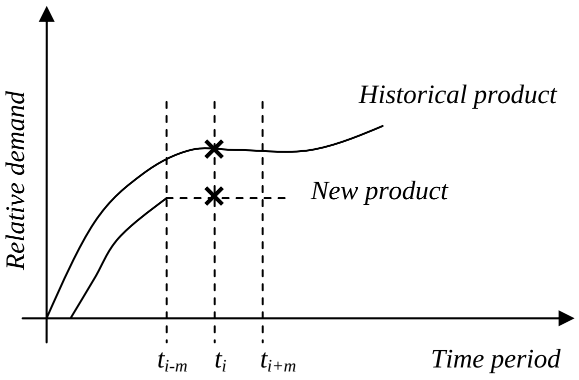


Figure 1: Three situations of transition time definition

The definition of the fact of transition can be conceptually described using three different techniques (see respective numbers in Fig. 1):

1. *Proactive approach* – we are at time moment t_{i-m} and wish to know the possibility that transition may occur at time t_i.
2. *Active approach* – we are at time t_i and wish to know the possibility that the transition is taking place right now.
3. *Reactive approach* – we are at time moment t_{i+m} and wish to know the possibility that the transition occurred at time t_i.

In a more general case, it is possible to predict (*proactive approach*), observe (*active approach*), or establish (*reactive approach*) the fact of transition both for one particular period, which corresponds to Fig. 1, and for a certain time interval.

From the point of view of practical evaluation of the above transition possibilities, it is evident that the proactive and active approaches will be characterised by higher uncertainty degree than the reactive approach. Parameter m must most probably be assumed to be equal to one, $m = 1$, which in combination with the reactive approach will enable creating an adequate model aimed to determine the fact of transition with the least uncertainty extent as compared to the two other approaches.

The situation with transition from maturity to the product's end-of-life is analysed similarly, which means that for such a transition the reactive approach is interpreted in the same way as for the transition from introduction to maturity.

2.1. Input historical data

The most practical and simplest case of available initial data is statistical data about the demand. Without doubt, in each period of time, in parallel with demand value, exogenous data exist, for example, orders and number of orderliness, number of customers for each product etc. Unfortunately, in practice it is difficult to obtain such complex information, therefore for the general case it is better to restrict data to the demand itself.

In general, the format of historical input data that could be processed by the system should (must) comply with these conditions:

- Each record displays the demand for a product, collected within known period of time, the length of which is set by the system – day, week, month, etc. In other words, each record is a demand time series.
- Each record has one or both markers – transition indicators:
  - Marker *K1* indicates the period when product switched from Introduction phase to Maturity phase;
  - Marker *K2* indicates the period when product switched from Maturity phase to End-of-Life phase.
- Each record has a marker, indicating the moment of the actual beginning of the Introduction phase (ABI).

The last condition is based on the fact that in many databases records are kept from the defined moment in time. It is evident that not all historical products were introduced on the market at the same moment in time. Marks on transitions can guarantee that a model will be build; if we have patterns of transitions in historical data, then, theoretically, in presence of a model for generalisation, we are able to recognise those patterns in new data.

3. MASTERING DATA: PRE-PROCESSING AND TRANSFORMATION

The main task of pre-processing demand data is to ensure that all time series are comparable. In the next turn, transformation is needed to remove excessive variety in the mentioned series, and also for representing initial task as a classification task. Practical utility of creating conditions for discovering generalities in time series is shown in the work (Das et al. 1998) where the authors propose to replace original time series with a sequence of discrete values, and after that to execute rule induction. The aim of the rule induction algorithm is to discover local regularities in that series. In case of recognition of transitions between phases, there is plenty of time series (that is why it is necessary to provide their comparability), and it is known that regularities should describe transitions exactly. Such specificity enables division into two classes – there is transition (*Yes*), there is no transition (*No*).

3.1. Pre-processing

Checking the fulfilment of the conditions mentioned in Section 2.1 is the first step in pre-processing historical input data. The next step in the pre-processing part is the shifting by *ABI*. The reason for this step to be included in the list is that *ABI* value varies between products. Shifting the records by the *ABI* corrects the variance of *ABI* between products by changing the period of actual beginning of Introduction to the first period. Together with *ABI* , the *K1* and the *K2* markers are shifted by the same number of periods as the *ABI* was – see the following formula (1).

$$\begin{aligned} K1' &= K1 - ABI + 1 \\ K2' &= K2 - ABI + 1 \end{aligned} \qquad (1)$$

After $K1'$ and $K2'$ are calculated, we can assign $K1'$ value to $K1$ and $K2'$ value to $K2$. The next important step after the *ABI* shifting is completed, is to select the proper records for the learning set. The main predefined parameters for selecting records are the minimal *K1* and the maximal *K2* transition periods. The minimal *K1* parameter or $K1_{\min}$ defines the minimal Introduction to Maturity phase transition period allowed to be passed to the system. The maximal *K2* parameter or $K2_{\max}$ defines the maximal Maturity to End-of-Life phase transition period allowed to be passed to the system. It should be noted, that for ensuring adequate results $K1_{\min}$ should be fixed and equal to three periods, $K1_{\min} = 3$.

Different learning datasets are formed for processing *K1* transition and processing *K2* transition. Due to the defined parameters, records with $K1 < K1_{\min}$ and records with $K2 > K2_{\max}$ are marked as non proper records and are not selected. If a record has both *K1* and *K2* markers, but only one of the markers does not fulfil the defined conditions, then the record still can be marked as a proper record, but only one of the transition markers will be used. This means that if *K1* does not match the conditions but *K2* does, then this record still can be used in learning set for processing the *K2* transition, and vice versa.

To compare different series and to mine for the knowledge from a dataset, the data normalization is needed. It is recommended to use the *Z*-score normalization, as it uses the standard deviation to normalize data, and the demand interval bounds are not used (Han and Kamber 2006). Normalization of each time series is made separately.

The next step after the normalization of the input data is the selection of necessary amount of demand values. From each proper *i*-th record the necessary amount of demand values will be taken – for *K1* processing the $[L_{K1}, U_{K1}]_i$ interval and for *K2* processing the $[L_{K2}, U_{K2}]_i$ interval.

Calculating the interval $[L_{K1}, U_{K1}]_i$ of necessary periods we have:

$$L_{K1,i} = 1. \qquad (3)$$

After shifting the records by ABI, the value of the $L_{K1,i}$ will be always equal to 1 – first period.

$$U_{K1,i} = \begin{cases} K1_i \; is\ odd : K1_i + 1 \\ K1_i \; is\ even : K1_i + 2 \end{cases}. \qquad (4)$$

Calculating the interval $[L_{K2}, U_{K2}]_i$ of necessary periods we get:

$$L_{K2,i} = K1_i + 1. \qquad (5)$$

Bound L_{K2} is the next period after the K1 switching period.

$$U_{K2,i} = \begin{cases} K2_i - L_{K2,i} + 1 \; is\ odd : K2_i + 1 \\ K2_i - L_{K2,i} + 1 \; is\ even : K2_i + 2 \end{cases}. \qquad (6)$$

As can be seen from equations (4) and (6) the right bounds of intervals are selected in such a way, that: first, to guarantee that there will be one demand value after transition point, and, second, that common number of values will be even. Even length of the series is a necessary condition for following transformation of the data.

3.2. Data discretization

On the basis of pre-processed set of historical data the so-called set of blocks is created. Blocks are formed using non-overlapping sliding window of two periods. The amount of usually supplied demand data is less than 2 years (24 months, or periods). Due to that, the length of the block equal to 2 periods is the most suitable way for data generalization, as it lessens the loss of the data. The set of blocks is formed by extracting the blocks from each record in the learning dataset.

In order to made discretization, it is necessary to discover clusters in the set of blocks. For clusterization of blocks it is enough to use a simple partitioning algorithm, such as k-means (Jain, Murty, and Flynn

1999). Indications for choosing the number of clusters k are considered in detail in Section 4.1. After clusters are found, time series are replaced with a sequence of numbers of clusters, representing corresponding blocks of values. For simplification of the following usage, a symbol, for example, “C” is placed before the number of cluster. In such a way, those data will be nominal.

3.3. Simulating online data

Carefully selected for the analysis and discretized historical data, as a matter of fact, represents terminal situation – after a definite number of periods transition to another phase of particular product life occurs. In the time of analysis of new products data will come *incrementally*, by one period, and within each period the system should analyze demand series and report to the user if there is transition or there is no. Taking into account the specificity of discretization, the analysis will be made after each two new values. In any case this process will take place *online*.

In order to guarantee stability of the system work with *online* data, it is necessary to simulate such data in the learning dataset. This process also allows creating two classes of events, there is transition (*Yes*), or there is no transition (*No*). For example, if a record, which contains transition *K1* in the historical data, after pre-processing and discretization consists of three blocks, then in the learning this record will be used twice. First time of two first blocks (since $K1_{\min} = 3$, then minimal analyzed number of blocks is two) and in associations with class ”*No*”. And also this record will be used a second time with all three blocks and in associations with class ”*Yes*”.

3.4. Transforming data

The transformation of the data is needed to come up with records of equal length. A special symbol “C0” is used to mark blocks without data available, and to come up with records of equal length. The new target attribute (class) “Transition” is added. This is a binary attribute containing “No” for records without transition (simulated data), and “Yes” for record with switching period – original full record. Attribute “Blocks” contains the number of blocks with data available.

Since maximal number of possible periods is determined by parameters of used database, then from this number follows maximal number of blocks and dimensionality of data table obtained after transformation. For example, if, as it was mentioned above, the database is restricted by 24 periods, it means that in total the table will consist of 13 attributes and the class. An example is given in Table 1. Before transformation, the indicated record with ID=1 and ID=2 was one record, which consisted of three blocks – see an example in Section 3.3.

Table 1: Example of transformed data, K1 transition

ID	Blocks	B1	B2	B3	…	B12	Transition
1	2	C3	C1	C0	…	C0	No
2	3	C3	C1	C2	…	C0	Yes

4. RECOGNITION MODEL

The specificity of the dataset obtained for creation of classifier is such, that there is expressed a class disbalance. The fact that the data describes real life process and marks of transitions were putted by experts implies that some noisiness in data is present. Also it is evident that such dataset can have a large size – more than a thousand records after transformation, even if the initial database included historical information on a few hundred products.

Which of known classifiers can be successfully applied to the above-mentioned data? Taking into account a possibility of transparent interpretation of the result, rule induction was selected, in particular, the RIPPER algorithm (an acronym for *repeated incremental pruning to produce error reduction*) (Cohen 1995). Classes (“Yes” and “No”) are examined in the increasing order and an initial set of rules for the class is generated on one set, and then pruned on a separate data set. Each rule is pruned immediately after it has been grown – it is incremental reduced-error pruning. Having produced a rule set for the class, each rule is reconsidered and two variants produced, again using reduced-error pruning – but at this stage, records covered by other rules for the class are removed from the pruning set, and success rate on the remaining instances is used as the pruning criterion. If one of the two variants yields a better description length, it replaces the rule. A final check is made to ensure that each rule contributes to the reduction of description length, before proceeding to generate rules for the next class.

For carrying out practical experiments, realization of this algorithm in the Weka environment (Witten and Frank 2005) was chosen. The name of this classifier in Weka is JRip. The main parameters are the amount of data used for pruning (one fold is used for pruning, the rest for growing the rules), and the number of optimization runs. In the conducted experiments these default values were used: folds=3 and optimizations=2.

4.1. Setting the number of clusters

The major part of all possible parameters is setup by the data. A question then arises regarding the selected number of clusters for data discretization (see also Section 3.2). From a theoretical point of view, a smaller number of clusters provides a greater generalization of demand information, but the increasing number of clusters gives an opportunity to describe different transition situations in more detail. It is evident that the balance between generalization and detailed elaboration should be found for each database individually.

If we divide learning data into a number of subsets and conduct cross-validation with different number of clusters, then the exact number of clusters for particular dataset can be found. Naturally, for this purpose the data on which clusters are discovered, are not used for testing obtained rules. The criterion for selecting the necessary number of clusters could be the recognition rate of class “Yes” in the tested data. If there is a possibility of setting costs of misclassifications of both classes “Yes” and “No”, then total cost of misclassification becomes the criterion. Total number or errors cannot be used as a criterion because there is a disbalance between classes.

5. EXPERIMENTS WITH K1 TRANSITION

In order to show, which rules can be found by the JRip algorithm, experiments on a real dataset with K1 marks and corresponding transitions were conducted. The original dataset contained information on about 235 products. After pre-processing, 199 records remained, and they were divided into three equivalent size subsets. Thus, the threefold cross-validation was performed.

In total, after transformation, 199 representatives of class “Yes” and 769 representatives of class “No” were obtained. Experiment has shown that if the number of clusters is smaller than 4 and greater than 10, the recognition level of class “Yes” noticeably falls. The number of found rules varied from 5 to 10. Below an example of obtained rules is given (on the one of validation cycles) when the number of clusters was 7:

- (Block3 = C6) and (N_blocks >= 4) => Class=Y (31.0/9.0)
- (N_blocks >= 9) and (Block6 = C6) => Class=Y (22.0/5.0)
- (Block3 = C6) and (Block2 = C2) => Class=Y (16.0/6.0)
- (N_blocks >= 5) and (Block4 = C6) => Class=Y (30.0/9.0)
- (Block10 = C3) => Class=Y (24.0/11.0)
- (Block4 = C3) and (Block5 = C3) => Class=Y (5.0/0.0)
- (Block2 = C3) and (N_blocks >= 6) => Class=Y (5.0/0.0)
- (Block2 = C6) and (N_blocks >= 3) => Class=Y (12.0/4.0)
- => Class=N (502.0/32.0)

The first eight rules describe class “Yes”, and the last rule indicates class “No”. After each rule, it is shown how many examples it covers and how many of these examples were misclassified. In this particular case recognition of class “Yes” was 71%, but recognition of class “No” 93%. It is evident, that obtained rules are easy to interpret and to explain to the business-users.

Is the obtained result occasional? To make sure that functioning of the system is stable, from all records in the original dataset (Set1) first values of demand were removed. The obtained set (Set2) was similarly tested at different numbers of clusters k, from 4 to 9. The received results for both datasets are shown in Figure 2. The recognition level of both classes for different datasets varied within 5%, which indicates the stable work of the developed system.

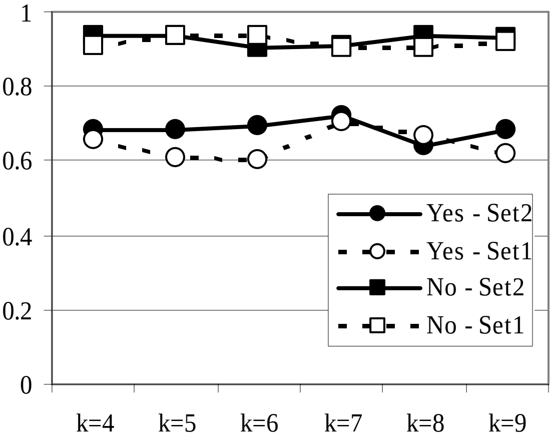


Figure 2: Recognition rate and used number of clusters

6. CONCLUSIONS AND FUTURE WORK DIRECTIONS

For the practitioners of management of the product life cycle the knowledge, which describes in which phase the product is, is topical. Such knowledge, in particular, helps to select between the cyclic and non-cyclic policy of planning supply chain operation.

In this paper, the task of recognition of transitions between different phases of product life is stated, and the structure of data mining system, which helps to solve this task, is shown. On the basis of the analysis of demand data on historical products it is possible to create stable classification rules, which are able to give online answers – if there is a transition in the particular product or there is no. The obtaining of such rules becomes possible thanks to creation of pre-processing and transformation system. The systems use demand time series as an input, and output strictly structured records. Since each obtained record has a class, it is possible to apply well- known and robust classifiers. From the point of view of implementation, the developed system is not complicated. All pre-processing and transformation processes are relatively simple, and it is possible to obtain rules in a well-implemented and available Weka environment.

One aspect is that, in the future it is necessary to examine the developed system on the data from different production fields, and, which is also important, to have a response from practitioners of supply chain management who will use these systems. In the near time described system will be implemented at a real enterprise.

Another aspect, modest data volume that was used for practical experiments, is related to the fact, that it is necessary to have transition marks in historical data from experts and practitioners. The more products, the more complicated for human to make all these marks – in practice the amount of marked data will always be restricted. As a result, possible direction of future research is treatment of recognition of transitions in the context of a semi-supervised learning (Zhu 2005). In this case, there is a small set with marked transitions (classes) and also a large dataset in which classes are not marked. In such a situation it is necessary to create a model, which will be able to recognize classes not only in the marked data, but also in the new (test) data.

ACKNOWLEDGMENTS

The present research was partly supported by the ECLIPS project of the European Commission "Extended collaborative integrated life cycle supply chain planning system", and also supported by ESF grant 2004/0002/VPD1/ESF/PIAA/04/NP/3.2.3.1/0001 /0002/0007.

REFERENCES

Campbell, G. M., and Mabert, V. A., 1991. Cyclical Schedules for Capacitated Lot Sizing with Dynamic Demands. *Management Science*, 37 (4), 409-427.

Cohen, W.W., 1995. Fast Effective Rule Induction. *Proceedings of the Twelfth International Conference on Machine Learning (ML95)*, pp. 115-123. (Tahoe City, CA).

Das, G., et al., 1998. Rule Discovery from Time Series. *Proceedings of the Fourth International Conference on Knowledge Discovery and Data Mining (KDD-98)*, pp. 16-22. (New York, USA).

Han, J., and Kamber, M., 2006. *Data Mining: Concepts and Techniques*. 2nd ed. San Francisco: Morgan Kaufman.

Jain, A.K., Murty, M.N., and Flynn, P.J., 1999. Data clustering: A review. *ACM Computing Surveys*, 31 (3), 264-323.

Kotler, P. and Armstrong, G., 2006. *Principles of Marketing*. 11th ed. Prentice Hall.

Merkuryev, Y., Merkuryeva, G., Desmet, B., and Jacquet-Lagrèze, E., 2007. Integrating Analytical and Simulation Techniques in Multi-Echelon Cyclic Planning. *Proceedings of the First Asia International Conference on Modelling and Simulation (AMS 2007)*, p. 460-464. (Phuket, Thailand).

Witten, I.H., and Frank, E., 2005. *Data Mining: Practical Machine Learning Tools and Techniques*. 2nd ed. Amsterdam: Morgan Kaufman.

Zhu, X., 2005 *Semi-supervised learning literature survey*. Technical Report 1530, Department of Computer Sciences, University of Wisconsin, Available from: http://pages.cs.wisc.edu/~jerryzhu/. [accessed on 10 February 2008]

EVOLUTIONARY METAMODELLING OF DISCRETE–EVENT SIMULATION MODELS

Birkan Can[(a)], **Gearoid Murphy**[(b)], **Cathal Heavey**[(c)]

[(a)] Enterprise Research Centre, University of Limerick
[(b)] Biocomputing and Developmental Systems, University of Limerick

[(a)] Birkan.Can@ul.ie, [(b)]Gearoid.Murphy@ul.ie, [(c)] Cathal.Heavey@ul.ie

ABSTRACT

In this work, we investigate evolutionary metamodelling of discrete-event simulation models with the buffer allocation problems. We propose a genetic programming approach in order to derive the artificial response functions of simulation models. Alternative to similar studies, we do not assume a form for the response function and perform symbolic regression analysis over simulation models of different sizes of serial production lines. We present a comparative analysis with another artificial technique, neural networks, to identify the efficiency and the performance of symbolic regression in deriving metamodels via simulation.

Keywords: simulation optimisation, metamodelling, genetic programming, symbolic regression, decision support, buffer allocation problem, neural networks.

1. INTRODUCTION

A system can be described as a mechanism embodying the relationships among the interdependent entities in order to perform certain objectives, accompanied by performance metrics. The study of interacting components and analysis of their contribution on the system performance requires descriptive tools. One such a tool, discrete-event simulation (DES), facilitates a flexible environment to model and investigate many systems dealt by Operational Research/Management Science. This way, it enables understanding of the system behaviour and its performance dependent on model configuration, which can be particularly useful in system design and optimisation.

A possible means of exploring the effects of the system designs on the performance is to apply trial-error approaches by manipulating the system parameters. However, many real-world problems are NP-complete, with a large number of alternative configurations (William, Jerzy and Bahill 2001). Therefore, 'what-if' type techniques may remain inefficient in identification of this effect (Simpson, Peplinski, Koch et al. 2001). Alternatively, metamodelling can be utilised to perform this task.

Furthermore, efficiency in obtaining the system performance is an important factor in analysis and improvement of systems. As introduced earlier, DES provides its user the ability of modelling a system in as much detail as desired. While this is an important aspect, model execution times can be long, e.g. for DES models of large and stochastic systems. Real world systems may be of large scale resulting in long simulation runs. Reflecting the stochastic nature of most DES models multiple simulation replications will be required. The efficiency in providing system performance appears to be particularly important in computation intensive tasks, e.g. design, optimisation (Li, Azarm, Farhang-Mehr et al. 2006). The required amount of simulation runs in such applications may further render exploration for better system options not only challenging, but also impractical. Alternatively, with some sacrifice from accuracy, metamodelling can be an appealing alternative in modelling the system behaviour to replace DES.

In this work, we will introduce an evolutionary approach, Genetic Programming (Koza 1992), for metamodelling of DES models. Genetic Programming (GP) is an evolutionary algorithm (EA) which has the ability to generate analytical models of the underlying training data via symbolic regression (SR). This opportunity is investigated first on different buffer allocation problems in serial production lines. Following, comparative analysis with Artificial Neural Networks (ANN) and complexity assessment of the results are presented to identify the performance and usability of SR in metamodelling of DES models.

The remainder of this paper is organized as follows. In the following section, the literature review accompanied with the brief description of the BAP is provided. Subsequently, related experimental work illustrating the efficiency of GP in response surface metamodelling of DES will be given in Section 3. Finally, discussion and future work conclude the article.

2. LITERATURE REVIEW

2.1. Buffer Allocation Problem

This study considers serial production lines (flow lines) of single machine servers, $\mathbf{M_i}$, and finite intermediate buffers, $\mathbf{B_i}$, of size $\mathbf{q_i}$, as shown in Figure 1, for m-station serial line. The machines are modelled to have exponential service rate with $\mu_i=1$ in order to reflect the stochastic nature of the manufacturing plant.

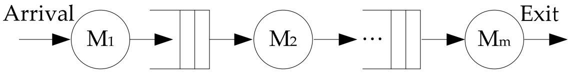


Figure 1: Serial production line.

The model of interest can be explained as follows. Jobs arrive at the system at the first buffer and sequentially proceed through the line. As a result of having finite buffers and variability in processing times, starvation and blocking can be observed at intermediate stages. When a machine is free to process, it takes a job from its upstream buffer. If there is no job to process in its upstream buffer, then the machine is said to be starved. Conversely, blocking in production lines can be commonly observed in two different ways, communication and production blocking (Papadopoulos, Heavey and Browne 1993). In our model, we assume that only production blocking occurs. In this type of blocking, a free machine takes a job from its upstream buffer for processing and passes it on to the downstream buffer after completion. However, if there is no space available in the buffer, the machine gets blocked and can not process new jobs. It is further assumed that the first machine is never starved and the last machine is never blocked.

Considering the illustrated processing scheme above, buffers can help improve efficiency and smooth operation of a manufacturing facility by eliminating disruptive effects of possible stochastic elements such as processing times and failures. The roles of buffers in manufacturing plants are further illustrated by Conway and Maxwell et al. (1988). The buffer allocation problem (BAP) addresses the efficient utilisation of storage spaces in manufacturing systems. It entails identifying the optimal allocation schemes to increase the performance of the operations. Besides manufacturing plants, similar problems can be observed in most supply chains operations and communication technologies (Dolgui, Eremeev and Sigaev 2007). Therefore, BAP has received a significant attention in research and practice, particularly in production and operations management.

The literature identifies the different aspects of production lines studied via buffer allocation. Kim and Lee (2001) investigate minimisation of work-in-process inventory (WIP) via buffer allocation, satisfying a minimum production rate (throughput rate) and an allowable buffer space. Similarly, Nahas and Ait-Kadi et al. (2006) and Diamantidis and Papadopoulos (2004) consider throughput rate as the performance criteria. Alternatively, Andijani and Anwarul (1997) evaluate the buffer allocation schemes via combining of WIP, cycle time and production rate. To obtain the performance data of a design, different tools such as; analytical queuing network models (De Almeida and Kellert 2000), Markov chain analysis (Vidalis, Papadopoulos and Heavey 2005) and simulation (Bulgak 2006) can be utilised. Analytical models are advantageous in the sense that system evaluations require less computation time. However, Altiparmak and Dengiz et al. (2007) indicate that they may suffer from being approximate models. Moreover, unrealistic assumptions may also be necessary to make the problem more tractable. An advantage of simulation over its analytical counterparts is its ability to reflect the system realities as much in detail as desired. However, its drawback appears in the time-cost of the evaluation of real-world systems as pointed out in Section 1. This may negatively effect the computational intensive applications involving the exploration of different alternatives, e.g. design, optimisation. To illustrate, despite the numerous optimisation techniques developed (Fu 2002, Tekin and Sabuncuoglu 2004), long simulation runs to evaluate the possible system alternatives may still render the overall process time-consuming. In such cases, approximate approaches can be alternatively used to estimate the response of a simulated system and to observe the effects of changes in model parameters. In the following section, a review of metamodelling in simulation is given.

2.2. Metamodelling in Simulation

It has been previously mentioned that metamodelling techniques can be appealing alternatives in modelling the system behaviour to replace DES with some sacrifice from accuracy when appropriate. There have been studies (Li, Azarm, Farhang-Mehr et al. 2006, Yang, Ankenman and Nelson 2007), in which simulation is coupled with metamodelling for more efficient performance measurements in the expense of the accuracy. Similarly, in this study a symbolic regression approach is taken in order to investigate metamodelling of the expected throughput rate dependent on the buffer allocation scheme.

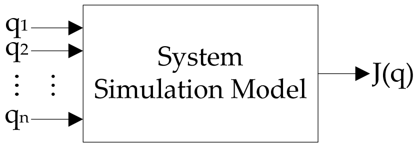


Figure 2: Simulation input-output relationship

A system, or particularly a simulation model, can be summarised as shown in Figure 2. Parameters; **q** is an input vector whose indices correspond to individual $\mathbf{q_i}$'s defining the decision variables of the system, where as $\mathbf{J_i(q)}$'s represent the performance criterion for a given configuration. In this work, **q** represents a buffer allocation scheme and expected throughput rate is the only performance measure considered.

Metamodelling can be defined as attaining an approximate function for $\mathbf{J_i(q)}$'s dependent on $\mathbf{q_i}$'s in order to understand behavioural aspects of the systems, e.g. estimation of the change in performance induced by the change in the configuration. It involves constructing approximate analytical models representing this function as in Equation 1, in which **q** is the system configuration and **ε** is the error in approximation.

$$\hat{J}(q) = f(q) \text{ such that } J(q) = \hat{J}(q) + \varepsilon \qquad (1)$$

The main concern in metamodelling is efficiently obtaining a reasonably accurate global model of the system (or a simulation model) (Wang and Shan 2007). While efficiency requires the minimum computational effort in attaining the metamodel, accuracy will reflect how close the metamodel is to the system. Generally, a metamodelling study is consisted of three major steps:

1. Sampling the data,
2. Modelling the collected data,
3. Fitting the model to sampled data.

Sampling refers to collecting sufficient data to reflect the system behaviour, i.e. attaining system performance at prespecified design points. Model choice and fitting are the subsequent steps in a metamodelling study. Briefly, the response is assumed to take a form, while this form can be a one or higher order polynomial dependent on the decision variables, network of neurons, decision trees or nonlinear transformations can be used as alternatives. There have been many techniques developed to perform these major steps above. Batmaz and Tunali (2002) give a comparative study considering sampling. Wang and Shan (2007) gives an extensive overview of the field outlining the roles and benefits of metamodelling. Simpson and Peplinski et. al (2001) present some of the popular methods, among which Polynomial Regression (PR), Kriging (KG), Artificial Neural Networks (ANN) have been prominently utilised methods in metamodelling DES. Jin, Chen and Simpson (2001) present a systematic comparison of common techniques on problems of different scales and difficulties.

There is a variety of metamodelling applications considering DES. In a recent study, Biles and Kleijnen et. al (2007) present a simulation-based optimisation study of constrained (s,S) inventory systems using KG. Noguera and Watson (2006) apply PR to attain the metamodel of a chemical plant as a function of capacity and throughput. Similarly, Durieux and Pierreval (2004) use second order polynomials to perform sensitivity analysis of a flexible manufacturing system to outline the effects of design parameters on average system utilisation. Afonin and Derjabkina et. al (2007) exemplify the use of ANNs in analysis of complex systems via metamodelling.

There have also been attempts exploiting ANNs to assist system evaluation in problems considering BAP. Artificial Neural Networks (ANNs), inspired by nature, emulate the nervous systems, such as human brain. As the interaction of neurons provides information processing capabilities, such as learning, ANNs can be trained over a set of data sample to generate the metamodels of simulated systems. Chambers and Mount-Campbell (2002) employ ANNs in a queuing system to derive product throughput rates and average system sojourn time to identify the optimum buffer sizes. Altiparmak, Dengiz and Bulgak (2007) investigate predictive capabilities of ANNs in assembly systems against polynomial and exponential regression models. Similarly, Bulgak (2006) uses ANNs in simulation metamodelling to obtain production rate of an assembly system in optimisation via buffer allocation. Finally, in a recent study, Person, Grimm and Ng (2008) exploits ANNs to assist optimisation routine in order to avoid long simulation runs.

The above discussions indicate that metamodelling can be employed as an efficient tool to obtain reasonably accurate global approximations to DES models. Such global models can contribute to understanding of the system behaviour based on the decision parameters. The literature review implies that there is still room for further research. In the following section, genetic programming will be introduced in order to perform based metamodelling of DES models based on symbolic regression.

2.3. Symbolic Regression with Genetic Programming

Evolutionary algorithms (EAs) are meta-heuristic techniques that are developed to solve difficult non-linear problems. In analogy to evolution in nature, EAs implement its operators and processes. The process of evolution imitated by EAs can be summarised as in the below pseudo-code:

```
1. Generate initial population, P(t);
2. Evaluate the individuals in P(t);
3. Repeat
      a. Select parents from P(t) to reproduce;
      b. Generate new offspring via crossover
         and mutation
      c. Evaluate the new individuals and
         insert to next generation;
4. Until termination.
```

The abstract definition of EAs as shown above has led to many sub-branches. Genetic Programming (GP) is an EA which can be used to evolve programs (Koza 1992). These programs may be interpreted as mathematical expressions, building instructions, grammar rules, etc. GP can generate symbolic representations of the training data which are often simply mathematical expressions, functions of the application domain.

Traditionally, regression refers to the use of numeric techniques to solve for the error against given training data, such as the back propagation algorithm of ANNs. Such approaches require prudent selection of configuration parameters to protect against the phenomenon of overfitting (Tetko, Alexander and Luik 1995), where the complexity of the model inhibits its performance on unseen data. Furthermore, the resulting set of numeric parameters are very difficult to interpret meaningfully for practitioners.

The use of GP for generating symbolic representations of training data was first proposed by (Koza 1992). Such representations are often simply mathematical expressions, functions of the application domain. Through the use of effective convergence algorithms, wherein the dynamics of evolution are enhanced via constraints, GP can produce compact expressions which have excellent generalisation properties (Murphy and Ryan 2008). These derived solutions have great potential for yielding insight into the underlying functional relationships of the application domain and have been used to aid geneticists in the comprehension of complex regulatory networks used in expressing DNA (Moore, Barney and White 2008).

A standard regression study involves determination of coefficients of a prespecified functional structure (generally a low order polynomial) which explains dependency of a parameter to independent design variables (Simpson, Peplinski, Koch et al. 2001). In contrast, symbolic regression does not require a strict specification of the size and shape, in other words, structural complexity of the solution. This stems from the fact that it requires a function set to evolve instead of assuming a compact analytical form. This implies that the functional form and its coefficients are explored simultaneously in symbolic regression to obtain response functions (Koza 1992).

This study uses these properties of GP to generate accurate symbolic models of DES which are potentially of great use in both optimising the processes and in enhancing the confidence practitioners have in using such artificially derived solutions by virtue of the succinct expressions produced by GP.

GP needs several important components to perform symbolic regression. GP uses a tree-based representation to search the space of solutions (see Figure 3.

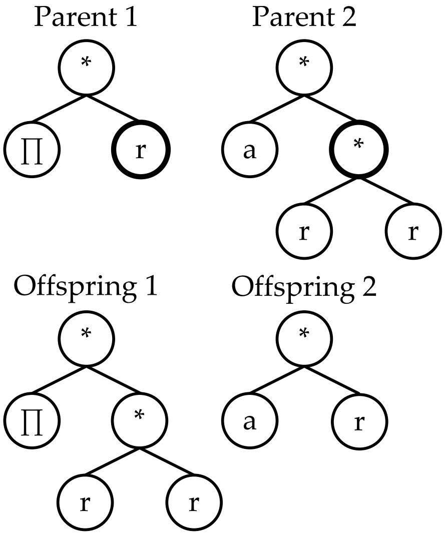


Figure 3: Tree-based representation and an example of crossover operation in GP. Crossover points on the parents are indicated by bold circles.

The traditional concepts of variation operators, i.e. recombination (crossover) and mutation in EAs, are adapted to the particular context of the GP representation, exemplified in the form of subtree swap in Figure 3. Similarly, mutation operates asexually and can perform removing (or replacing) subtrees. A discussion on the role of variation operators can be found in Luke and Spector (1997). Further details regarding the GP implementation can be found in Section 3.

There have been a variety of applications of symbolic regression via genetic programming reported in the literature. Duffy and Warnick (1999) highlight the ability of GP in deriving functions implying strategies in decision-making. Castillo and Kordon et. al (2005) apply symbolic regression for statistical model building in industrial chemical processes. In another application, Korns (2007) investigates symbolic regression on large-scale complex problems and tries improving the efficiency of SR via integration of formal grammar rules to modelling process.

To conclude, the above discussions justify the use of GP in simulation-based metamodelling. The following section will present the experiments to assess the quality and usability of symbolic regression in metamodelling of DES.

3. EXPERIMENTS AND RESULTS

This section presents the results of a set of experiments in which the ability of GP to generate accurate metamodels of simulations is compared and contrasted against an ANN performing the same task. Both systems used the buffer size of the stations in the simulation as their input and attempted to predict the throughput rate of the simulation. The performance graphs show the testing performance of the systems, a good measure of generalisation ability. The performance of the techniques are represented via root mean squared error between the predicted throughput and the actual throughput as 1/(1+RMSE). All results shown here are the results of 100 independent runs.

The testing and training data was derived by generating a random distribution of buffer sizes amongst the stations of the model and evaluating the resulting throughput rate for that buffer configuration. Two sets of 100 buffer samples were used for testing and training on each simulation model. We used simulation models consisting of 4, 8, 12, 16 and 20 stations with a maximum buffer size of 20.

The GP algorithm used here employs a variation of the Hereditary Repulsion (HR) convergence manipulation protocol to enable the GP evolve compact and powerful expressions (Murphy and Ryan 2008). The algorithm uses a generational framework in which the next generation is produced entirely from the current generation. Crossover events cannot span more than one generation. The initial population is created using the ramped half and half initialisation method (Koza 1992). This produces a broad spread of expressions of varying size. Each expression is given a fitness derived from the RMSE of its ability to predict throughput; 1/(1+RMSE).

Random selection is used to pick parents from the population. This selection strategy increases performance by reducing the pressure on the population to converge, thus allowing evolution more time to discover useful functionality. Once an offspring from the random parents has been evaluated, it is only allowed into the next generation if it is better than *both* parents. Should the offspring fail this criterion, it is discarded and the process begins again. It is not possible for a parent to move from the current generation to the next generation. It must preserve its genes by combining with another solution in a manner superior to both parents.

Because of the potential for failed crossovers, the number of evaluations per generation is variable with the HR algorithm; therefore it must be compared to other systems in terms of evaluations. The functional primitives used in the GP algorithm are as follows; (*, %, +, -, exponent, sqrt, cos, sin). Subtree crossover was used without any mutation, as the HR algorithm does not require it. The parameters used to initialise the GP settings are described in Table 1.

Table 1: GP Parameters

Parameters	Value
Population Size	1000
Min Initial Tree Size	2
Max Initial Tree Size	4
Max Tree Size	12
Initialisation Method	Ramped Half and Half
Evaluation Limit	10^6

The Multi-Layer Perceptron (MLP) used a standard back-propagation algorithm to perform learning (Rumelhart, Hinton and Williams 1986). Its parameters are described Table 2.

Table 2: MLP Parameters. These parameters were found to consistently give the best MLP performance across all the experiments.

Parameters	Value
Hidden Layer Size	10
Learning rate	0.01
Initialisation Range	+/-1

Table 3: GP vs. ANN testing fitness results giving a statistical confidence of %99.9 on GP performance surpasses the ANN for the given configurations.

Simulation	GP Fit	T-Value	MLP Fit
4 Station	0.944215	-2.236917	0.945442
8 Station	0.939218	30.427879	0.928591
12 Station	0.941697	52.115238	0.933142
16 Station	0.943745	22.775646	0.939181
20 Station	0.947534	29.964231	0.941968

The results are tabulated in Table 3. The Students T-Test was evaluated for each experiment. Table 3 clearly shows that GP outperforms the MLP for all experiments except the easiest 4 station configuration. In absolute terms however; the performance difference between the paradigms is slight. Despite the slim difference between GP and the MLP, it was observed consistently in the experiments and produced high statistical confidence values.

A figure illustrating the regression performance for both GP and the MLP over a duration of 10^6 evaluations is given in Figure 4. This shows that the MLP quickly converges but fails to progress any further. By contrast, GP is slow to converge but manages to successfully sustain convergence, finding higher quality solutions than the MLP.

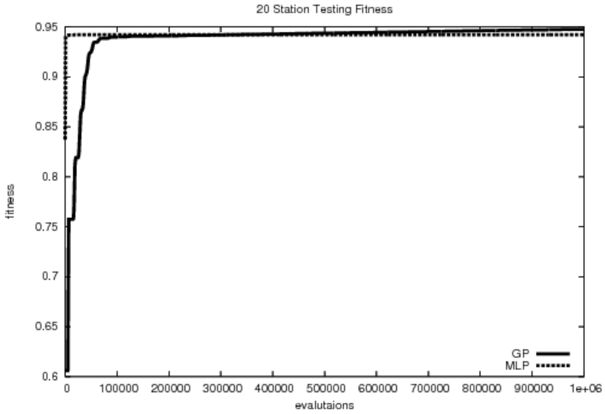


Figure 4: Graph illustrating contrasting the regression dynamics of MLP and GP for the 20 station problem. MLP quickly converges but GP eventually surpasses the performance of the MLP.

A clear sense of the difference between the GP and MLP emerges when one examines the complexity of the resulting solutions. We measure its complexity as the number of components involved in contributing to the solution. For the MLP, this is the number of weights in the network. For the 4 station experiment the MLP consisted of 50 weights, 4 input nodes by 10 hidden nodes plus the final 10 weights for the output nodes. Similarly, for the 8, 12, 16 and 20 station experiments, the MLP consisted of 90, 130, 170 and 210 components.

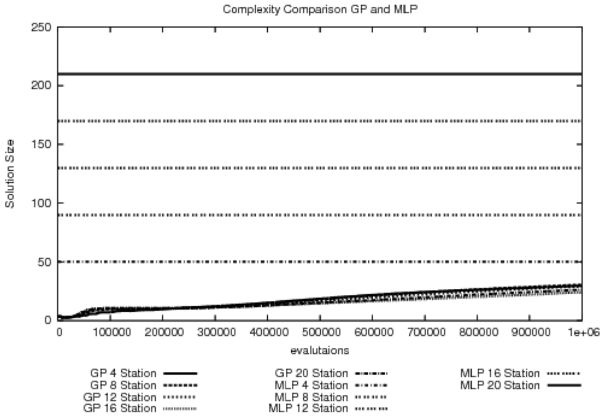


Figure 5: Fixed size MLP solutions are far more complex than the GP solutions.

By contrast, GP consistently utilises a small fraction of the components used by the MLP, illustrated in Figure 5. These smaller compact solutions immediately provide a viable platform for analysing the dynamics of the simulation model. Table 4 shows in detail the actual average sizes of evolved solutions in the GP population.

Table 4: Table illustrating the contrast in size of solutions between GP and MLP.

Simulation	GP Tree Node Size	MLP Weights Size
4 Station	29.6925	50
8 Station	30.5628	90
12 Station	29.0962	130
16 Station	24.0059	170
20 Station	26.1847	210

4. DISCUSSION AND CONCLUSIONS

Discrete-event simulation provides a flexible modelling platform enabling the detailed analysis of many industrial systems. However, long model execution time particularly is a common challenge in simulation-based computation intensive studies, such as design and optimisation. In this work, we have tried to address this issue with symbolic regression via genetic programming (GP) by generating analytical approximations with %0.06 proximity to the actual model behaviour.

In order to assess the applicability of genetic programming in metamodelling of simulation, serial production lines with buffer allocation problem were studied to identify dependency of average expected throughput rate on the individual buffer allocations. We have compared our results against ANNs, which are prominently applied in metamodeling of buffer allocation problem. The two techniques were analysed in terms of the quality of approximation and complexity of the generated models. It has been statistically shown that GP performance is not worse than the implemented ANN for the given test problems in terms of the accuracy of the approximation. Moreover, GP has demonstrated a tendency towards surpassing the performance of the ANN. We believe, this ability can be further improved with the use of appropriate sampling techniques to attain the training data, whereas randomly selected design points are utilised in our experiments.

Furthermore, a clear sense of the difference between GP and ANN emerges when the complexity of the resulting solutions are examined. Our results have illustrated that GP consistently utilises a small fraction of the components used by the ANN. These smaller compact solutions immediately provide a viable platform for analysing the dynamics of the simulation model. This fact implies a considerable use to the practitioner as GP has provided a tangible functional decomposition of the simulation. In this respect, symbolic regression via GP can allow a window into the `black box' of the simulation.

GP has been observed to exhibit latency at the initial stage of the experiments which is a consequence of the system expending a great effort to improve the initial random content of the population. However, this latency is a negligible drawback considering the fact that time cost of obtaining a metamodel is in fractions of attaining training data. Nonetheless, we would like to focus on this value loss in our future studies.

Finally, an interesting observation deserves to be pointed out in model complexity. The complexity analyses indicate that GP consistently produces smaller solutions despite the problem size getting larger. This may be due to the fact that effect of certain buffer positions in the serial line may be more significant than others in describing the throughput rate of the model. While this leads GP to generate less complex expressions, it actually locates the bottleneck positions through the production line as well. This observation will need further analysis in future work.

ACKNOWLEDGMENTS

This research is funded by IRCSET, Irish Research Council for Science, Engineering and Technology.

REFERENCES

Afonin, P., V. Derjabkina, A. Kozhukhova and O. Lamskova (2007). "The design and analysis of complex system neural network metamodels." *Optical Memory \& Neural Networks* **16**(3): 154-158.

Altiparmak, F., B. Dengiz and A. A. Bulgak (2007). "Buffer allocation and performance modeling in asynchronous assembly system operations: An artificial neural network metamodeling approach." *Applied Soft Computing* **7**(3): 946-956.

Andijani, A. A. and M. Anwarul (1997). "Manufacturing blocking discipline: A multi-criterion approach for buffer allocations." *International Journal of Production Economics* **51**(3): 155-163.

Batmaz, I. and S. Tunali (2002). "Second-order experimental designs for simulation metamodeling." *Simulation-Transactions of the Society for Modeling and Simulation International* **78**(12): 699-715.

Biles, W. E., J. P. C. Kleijnen, W. C. M. van Beers and I. A. van Nieuwenhuyse (2007). Kriging metamodeling in constrained simulation optimization: an explorative study

Kriging metamodeling in constrained simulation optimization: an explorative study. *Simulation Conference, 2007 Winter*.

Bulgak, A. A. (2006). "Analysis and design of split and merge unpaced assembly systems by metamodelling and stochastic search." *International Journal of Production Research* **44**: 4067-4080.

Castillo, F., A. Kordon, J. Sweeney and W. Zirk (2005). Using Genetic Programming in Industrial Statistical Model Building. Genetic

Programming Theory and Practice II, Springer US**:** 31-48.

Chambers, M. and C. A. Mount-Campbell (2002). "Process optimization via neural network metamodeling." *International Journal of Production Economics* **79**(2): 93-100.

Conway, R., W. Maxwell, J. O. McClain and L. J. Thomas (1988). "The Role of Work-in Progress Inventory in Serial Production Lines." *Operations Research* **36**(2): 229.

De Almeida, D. and P. Kellert (2000). "An Analytical Queueing Network Model for Flexible Manufacturing Systems with a Discrete Handling Device and Transfer Blockings." *International Journal of Flexible Manufacturing Systems* **12**(1): 25-57.

Diamantidis, A. C. and C. T. Papadopoulos (2004). "A dynamic programming algorithm for the buffer allocation problem in homogeneous asymptotically reliable serial production lines." *Mathematical Problems in Engineering* **2004**(3): 209-223.

Dolgui, A., A. Eremeev and V. Sigaev (2007). "HBBA: hybrid algorithm for buffer allocation in tandem production lines." *Journal of Intelligent Manufacturing* **18**(3): 411-420.

Duffy, J. and J. Warnick (1999). Using Symbolic Regression to Infer Strategies from Experimental Data, Society for Computational Economics.

Durieux, S. and H. Pierreval (2004). "Regression metamodeling for the design of automated manufacturing system composed of parallel machines sharing a material handling resource." *International Journal of Production Economics* **89**(1): 21-30.

Fu, M. C. (2002). Optimization for discrete-event simulation: Theory and practice. System Simulation and Scientific Computing. Z. J. Chen. Beijing, Int Acad Publ Beijing World Publ Corp**:** 50-50.

Jin, R., W. Chen and T. W. Simpson (2001). "Comparative studies of metamodelling techniques under multiple modelling criteria." *Structural and Multidisciplinary Optimization* **23**(1): 1-13.

Kim, S. and H. J. Lee (2001). "Allocation of buffer capacity to minimize average work-in-process." *Production Planning and Control* **12**: 706-716.

Korns, M. (2007). Large-Scale, Time-Constrained Symbolic Regression. Genetic Programming Theory and Practice IV**:** 299-314.

Koza, J. R. (1992). *Genetic Programming: On the Programming of Computers by Means of Natural Selection*. Massachusetts, The MIT Press.

Li, G., S. Azarm, A. Farhang-Mehr and A. R. Diaz (2006). "Approximation of multiresponse deterministic engineering simulations: a dependent metamodeling approach." *Structural and Multidisciplinary Optimization* **31**(4): 260-269.

Luke, S. and L. Spector (1997). A Comparison of Crossover and Mutation in Genetic Programming. *Genetic Programming 1997: Proceedings of the Second Annual Conference*, Stanford University, CA, USA.

Moore, J. H., N. Barney and B. C. White (2008). Solving Complex Problems in Human Genetics Using Genetic Programming: The Importance of Theorist-Practitionercomputer Interaction. Genetic Programming Theory and Practice V, Springer Berlin / Heidelberg**:** 69-85.

Murphy, G. and C. Ryan (2008). A Simple Powerful Constraint for Genetic Programming. *Genetic Programming,* Springer Berlin / Heidelberg**:** 146-157.

Nahas, N., D. Ait-Kadi and M. Nourelfath (2006). "A new approach for buffer allocation in unreliable production lines." *International Journal of Production Economics* **103**(2): 873-881.

Noguera, J. H. and E. F. Watson (2006). "Response surface analysis of a multi-product batch processing facility using a simulation metamodel." *International Journal of Production Economics* **102**(2): 333-343.

Papadopoulos, H. T., C. Heavey and J. Browne (1993). *Queuing Theory*, Springer.

Person, A., H. Grimm and A. Ng (2008). A Metamodel-Assisted Steady-State Evolution Strategy for Simulation-Based Optimization. *Trends in Intelligent Systems and Computer Engineering:* 1-13.

Rumelhart, D. E., G. E. Hinton and R. J. Williams (1986). "Learning representations by back-propagating errors." *Nature* **323**(6088): 533-536.

Simpson, T. W., J. D. Peplinski, P. N. Koch and J. K. Allen (2001). "Metamodels for computer-based engineering design: survey and recommendations." *Engineering with Computers* **17**(2): 129-150.

Tekin, E. and I. Sabuncuoglu (2004). "Simulation optimization: A comprehensive review on theory and applications." *IIE Transactions* **36**(11): 1067-1081.

Tetko, I. V., D. J. L. Alexander and I. Luik (1995). "Neural Network Studies. 1. Comparison of Overfitting and Overtraining." *Chemical Information and Modeling* **35**(5): 826-833.

Vidalis, M. I., C. T. Papadopoulos and C. Heavey (2005). "On the workload and 'phaseload' allocation problems of short reliable production lines with finite buffers." *Comput. Ind. Eng.* **48**(4): 825-837.

Wang, G. G. and S. Shan (2007). "Review of metamodeling techniques in support of engineering design optimization." *Journal of Mechanical Design* **129**(4): 370-380.

William, L. C., R. Jerzy and A. T. Bahill (2001). "System design is an NP-complete problem: Correspondence." *Syst. Eng.* **4**(3): 222-229.

Yang, F., B. Ankenman and B. L. Nelson (2007). "Efficient generation of cycle time-throughput curves through simulation and metamodeling." *Naval Research Logistics* **54**(1): 78-93.

DATA MINING VIA DISTRIBUTED GENETIC PROGRAMMING AGENTS

Gabriel K. Kronberger[(a)], Stephan M. Winkler[(b)], Michael Affenzeller[(c)], Stefan Wagner[(d)]

[(a) (b) (c) (d)]Heuristic and Evolutionary Algorithms Laboratory
School of Informatics, Communications and Media
Upper Austria University of Applied Sciences
Softwarepark 11, 4232 Hagenberg, Austria

[(a)]gkronber@heuristiclab.com, [(b)]swinkler@heuristiclab.com, [(c)]maffenze@heuristiclab.com, [(c)]swagner@heuristiclab.com

ABSTRACT

Genetic programming is a powerful search method which can be applied to the typical data mining task of finding hidden relations in datasets. We describe the architecture of a distributed data mining system in which genetic programming agents create a large amount of structurally different models which are stored in a model database. A search engine for models that is connected to this database allows interactive exploration and analysis of models, and composition of simple models to hierarchical models. The search engine is the crucial component of the system in the sense that it supports knowledge discovery and paves the way for the goal of finding interesting hidden causal relations.

Keywords: distributed data mining, genetic programming, knowledge discovery

1. INTRODUCTION

1.1. Data Mining and Genetic Programming

Hand et al. give the following definition of data mining: *"Data mining is the analysis of (often large) observational data sets to find unsuspected relationships and to summarize the data in novel ways that are both understandable and useful to the data owner."* (Hand, Mannila and Smyth 2001). Data mining is just one step in the more comprehensive process of knowledge discovery. Other equally important steps include the preparation of data for the mining process and subsequent interpretation of generated models; cf. (Fayyad, Piatetsky-Shapiro and Smyth 1996). The goal of the whole process is to gain new knowledge about the observed system which can be utilized consequently to improve aspects of the system, for instance to gain a competitive advantage.

Genetic programming is an optimization technique that works by imitating aspects of natural evolution to generate a solution that maximizes or minimizes a fitness function. A population of solution candidates evolves through many generations towards a solution using three evolutionary operators: selection, recombination and mutation. Genetic programming is based on genetic algorithms the main difference is the representation of solution candidates, whereas genetic algorithms are intended to find an array of characters or integers representing the solution of a given problem, the goal of GP is to produce a computer program solving the problem at hand.

Genetic programming has been used successfully for data mining tasks using different forms of solution representations. One approach is symbolic regression to build formulas that describe the behavior of systems from measured data, see for example (Koza 1992; Keijzer and Babovic 1999; Langdon and Poli 2002; del Re, Langthaler, Furtmüller, Winkler and Affenzeller 2005; Winkler, Affenzeller and Wagner 2007a; Winkler, Affenzeller and Wagner 2007b). Other approaches use GP to discover predictive IF-THEN-rules typically with prior discretization of variables (Freitas 1999; Wong and Leung 2000) or to evolve decision trees (Fu 1999; Ryan and Rayward-Smith 1998; Papagelis and Kalles 2001).

In this paper we describe a system to support and improve the knowledge discovery process based on distributed data mining agents which concurrently run genetic programming processes. In section 2 we describe the main problems of GP-based data mining and describe potential benefits of a distributed data mining system. In section 3 we describe the architecture of this system and its major components while sections 4 and 5 conclude this paper with ideas to further improve the knowledge discovery process by integrating a priori knowledge and user feedback.

2. MOTIVATION

A frustrating aspect of GP is that it takes a long time until the result of a run is available. Especially for non-trivial datasets it is usually necessary to analyze the result of a previous run before a new run can be started for instance to counteract over or under fitting or to exclude dominant input variables. This is an even bigger problem for domain experts who do not fully understand the internal details of GP and thus often have problems to configure the algorithm correctly. Usually a few iterations are necessary until a configuration for the algorithm is found that works for data mining task at hand. However, even when such a configuration has been found there is another aspect that causes friction in the knowledge discovery process.

Multiple genetic programming runs with the same settings and the same input data typically result in a diverse set of structurally different models with similar predictive accuracy. This behavior is caused by the fact that GP is a stochastic method which searches for a model that fits the target variable as well as possible. GP specifically doesn't search for the simplest or most compact model, on the contrary internal dynamics of the evolutionary process cause the models to become overly complex. This effect is known in the GP community as “bloat”; cf. (Langdon and Poli 2002). Various strategies to combat bloat have been discussed in GP literature, one recent addition is (Dignum and Poli 2008), giving a full overview would go beyond the scope of this paper. For the task at hand the effect can be alleviated through simplification of the resulting models. However the basic problem remains because there are often implicit dependencies between input variables and there is usually an infinite number of ways to express the same function.

The result is that it is difficult to extract knowledge out of the generated models because the relevant information about the underlying structure in the data is blurred by the large amount of possible mathematical representations of that structure. The knowledge gained from these experiments is often limited to the insight into which variables play an important role in models for the target variable. While this insight can be valuable in itself it can also be reached with statistical methods with a lot less effort. One important feature of GP that distinguishes it from many other optimization methods is that it is able to automatically optimize the model structure while at the same time optimizing the model parameters. This characteristic cannot be utilized to its full extent when the analyzed models are all structurally different and thus difficult to analyze which thwarts the effort spent to build the model structure.

One approach to improve the discovery of more detailed knowledge is to run many independent GP processes to generate a large number of models for each possible target variable with different complexities and to extend the data mining process to search for implicit dependencies between input variables. In combination with an interactive user-interface to filter and analyze the generated models and to compose simple models to hierarchic models the user gains new knowledge step by step while investigating the set of models.

Evolutionary algorithms especially genetic programming are often slower than other more specialized data mining algorithms while reaching comparable predictive accuracy. However it's easy to parallelize evolutionary algorithms. In the proposed data mining system this is even simpler since the independent genetic programming processes can be executed concurrently. Depending on the complexity and extent of the dataset it can still take hours or even a few days to generate enough models to start interpreting the generated models, however as Freitas states: *“Data mining is typically an off-line task, and in general the time spent with running a data mining algorithm is a small fraction (less than 20%) of the total time spent with the entire knowledge discovery process. [...] Hence, in many applications, even if a data mining algorithm is run for several hours or several days, this can be considered an acceptable processing time, at least in the sense that it is not the bottleneck of the knowledge discovery process.”* (Freitas 2002). A benefit of the system is that the user can already start to analyze preliminary results while the background GP processes are still refining models.

The design goals of the system can be summarized in the following four points:

- Find all potentially interesting (non-linear) relations of variables of a dataset.
- Store models of different complexity and accuracy.
- Provide functionality to explore, analyze and compose such models.
- Record all steps in the mining process which led to a given result.

These goals are closely related to the goals given by Blumenstock, Schweiggert and Müller 2007 in that they result out of similar considerations regarding the focus and breadth of the search process and the transparency and ease of use of the system.

In the following section we describe the architecture of the proposed system and its components.

3. DISTRIBUTED DATA MINING ARCHITECTURE

The system is made up of three parts: a central model database, a mesh of distributed genetic programming agents and a component to import new datasets into the system and to navigate, explore and search models stored in the central database.

3.1. Layout of the Generic Model Database

Figure 1 shows an entity-relationship diagram of the generic data model of the model database. The two most important entities in the data model are the dataset and the model. It is often the case that a dataset once imported is preprocessed for instance to scale all variable values to a predefined range. For sake of transparency each dataset is linked back to its source. Later all processing steps can be retraced with this relation. Additionally the person who imported or manipulated the dataset is linked to each dataset. Each model is linked to the process that generated it to make the origin of that model transparent. Each process is also linked back to a person who is the controller of that process. By adding the algorithm that each process executes in the data model it is possible to repeat each experiment at a later point in time.

The layout of the data model is kept very generic on purpose to make it easy to add new data mining algorithms with different model representations to the system. Algorithms could be different genetic programming variants or even other (non-evolutionary) data mining algorithms like C4.5, kNN, CART or SVM.

The main design goal of the model database is transparency. It must be possible to reproduce every single result that is stored in the database. For this reason the algorithm implementation and the parameter settings are also stored in the database.

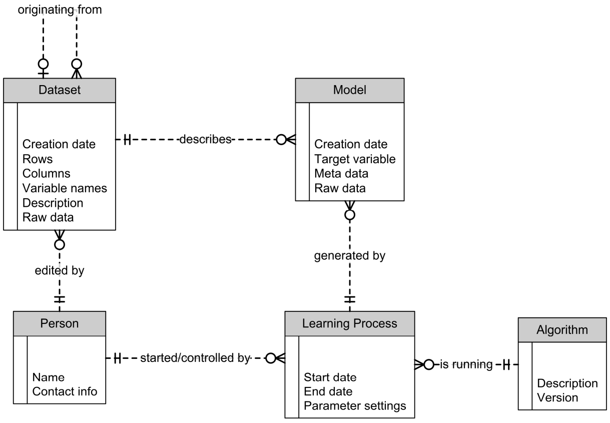


Figure 1: Entity-relationship diagram showing the generic data model of the model database.

3.2. Distributed Genetic Programming Agents

Figure 2 shows the typical cycle of data mining with genetic programming. The user supplies the dataset and configures the parameters of the algorithm. The most important parameters are the set of allowed input variables the target variable and the set of functions that should be used to compose the models. After a few hours the result of the algorithm in the form of a formula is available. Through the analysis of this result the user gains new knowledge and starts a new GP run with different settings (for instance removing an input variable).

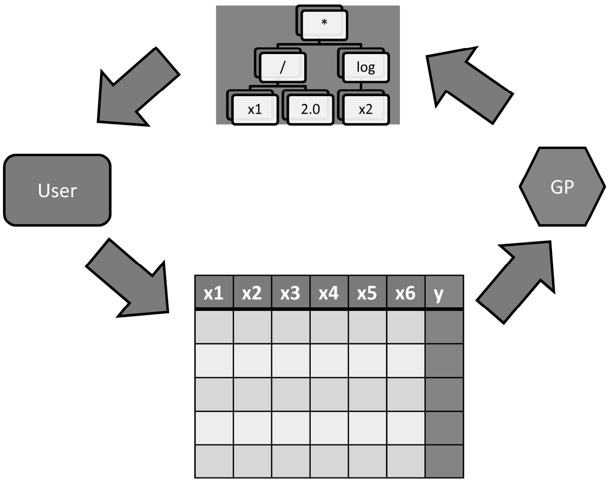


Figure 2: The usual way of GP-based data mining has a cycle with a long feedback loop.

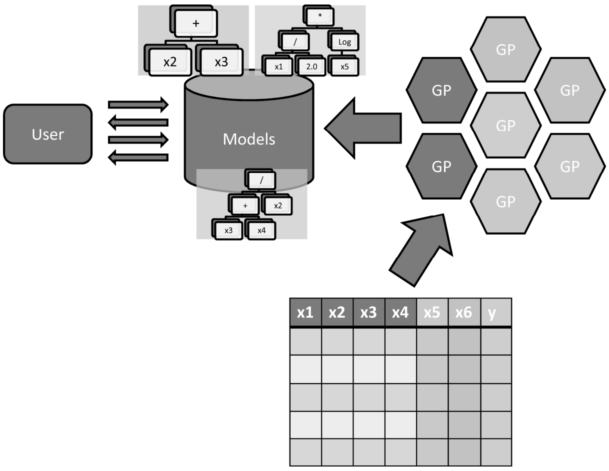


Figure 3: In the proposed system distributed GP agents continuously analyze the dataset and store new models in the database. The user interactively explores and analyzes available models.

Figure 3 shows how this process could be improved by using parallelism to run different GP processes at the same time. Each of the distributed genetic programming agents has different settings for the target variable, maximally allowed model complexity and the set of allowed input variables. Controller agents create new GP jobs and coordinate the running GP agents.

It would be interesting to have more intelligent controller agents which try to predict which models are more interesting for the user and guide the GP agents to search especially for such models; this remains a topic for future research.

3.3. Interactive Model Exploration and Analysis

The mesh of distributed genetic programming agents generates a flood of potentially interesting models. However only a few of these models will actually be interesting for the user and it is usually impossible to automatically recognize the relevant models and throw out the rest. So an user-interface for the interactive exploration and analysis of all available models is essential. It is the most crucial component for the knowledge discovery process because the facts the user is searching for are likely hidden and can only be uncovered when it is possible to arbitrarily filter and sort the available models and drill down to uncover alternative representations of a model.

The quality of any model can be inspected visually through line charts of the estimated and the original value of the target variable and through scatter plots showing the correlations of estimated vs. original values. The relative complexity of models can be visualized through different colors.

We plan to implement a kind of search engine for models which allows filtering and sorting available models for a given dataset by at least the following attributes:

- Target variable
- Input variables
- Accuracy
- Complexity

Search queries can be freely combined and negated for instance to create the search request *"Find models for variable X which do not contain variable Y and Z sorted by accuracy"* or *"Find models for variable Z using variable A and B with a maximal tree-depth of four"*. Additionally to the basic filtering and sorting functionality it is interesting to explicitly search for similar models to a given model. This can be implemented by searching for models with the same target variable and the same input variables. Searching for structural similarity could also be useful but remains an open topic.

To make the search process transparent to the user it will be possible to display which algorithm in combination with which settings produced a given model. Going a step further it's also interesting to show the internal state of the algorithm when it produced the given model. This information is especially helpful to refine and improve the distributed data mining system itself.

4. FUTURE WORK

The first step towards an interactive data mining environment for practical application is the implementation and roll-out of the system described in the previous sections. We plan to combine the distributed data mining system with HeuristicLab (available at http://www.heuristiclab.com) (Wagner, Affenzeller 2005), a modern framework for prototyping and analyzing optimization techniques for which both generic concepts of evolutionary algorithms and many functions to evaluate and analyze algorithms are available. HeuristicLab makes it very easy to create customized algorithms from predefined components. It is a close to ideal environment for the data mining system because as Hand et al. stated: "*When faced with a data mining application, a data miner should think about which* components *fit the specifics of his or her problem, rather than which specific "off-the-shelf" algorithm to choose. In an ideal world, the data miners would have available a software environment within which they could* compose *components (from a library of model structures, score functions, search methods, etc.) to synthesize an algorithm customized for their specific applications. Unfortunately this remains a ideal state of affairs rather than the practical norm; current data analysis software packages often provide only a list of algorithms, rather than a component-based toolbox for algorithm synthesis."* (Hand, Mannila and Smyth 2001).

Once the basic infrastructure is implemented and running there are a few more possible research directions additionally to the open topics mentioned above.

One interesting aspect is to enhance the cooperation between the distributed GP agents by reusing models from the model database generated by other agents. These models could be integrated into other models as virtual variables replacing the actual training data with predictive models for those variables. This scheme enhances reuse of learned concepts and knowledge and enables the creation and composition of successively higher-level models. This idea is related to automatically defined functions (Koza 1992) and similar to adaptive representation through learning (Rosca and Ballard 1996).

A possible direction for future research is extending the user-interface to automatically learn and recognize which models are interesting for the user. One possible approach is to track user search actions and using a form of online data mining to analyze and predict the interestingness of a given model based on previous user actions.

5. SUMMARY

In this paper we describe the architecture of a distributed data mining system utilizing genetic programming agents to compose a diverse set of predictive models for a given dataset. The system is made up of three parts: a central model database, a mesh of distributed genetic programming agents and a component to import new datasets into the system and to navigate, explore and search models stored in the central database.

In the proposed system distributed genetic programming agents continuously process available datasets and create models of different complexity for different target variables and for different sets of input variables. These agents store potentially interesting models in a database together with meta-data to filter and sort models. Because of the the high volume and diversity of the model database an interactive user-interface to navigate, filter and analyze the models is necessary. This interface is the crucial component in the knowledge discovery process in the sense that it allows interactive exploration of results and paves the way for the goal of finding interesting hidden causal relations.

ACKNOWLEDGMENTS

The research presented in this paper is funded by the Upper Austrian University of Applied Sciences in the scope of the project "Cooperative Evolutionary Data Mining Agents".

REFERENCES

Blumenstock, A., Schweiggert F., Muller M., 2007. Rule cubes for causal investigations. *Proceedings of the 7th IEEE International Conference on Data Mining (ICDM 2007)*, pp. 53 – 62. October 28-31 2007, Omaha, Nebraska, USA.

Del Re, L., Langthaler, P., Furtmüller, C., Winkler, S., Affenzeller, M., 2005. NOx Virtual Sensor Based on Structure Identification and Global Optimization. *Proceedings of the SAE World*

Congress 2005, paper number 2005-01-0050. April 11-14 2005, Detroit, MI, USA.

Dignum, S., Poli, R., 2008. Operator equalisation and bloat free GP. *Genetic Programming, 11th European Conference, EuroGP 2008*, pp. 110 – 121. March 26-28 2008, Naples, Italy.

Fayyad, U. M., Piatetsky-Shapiro, G., Smyth, P., 1996. From data mining to knowledge discovery: an overview. In Fayyad, U. M., Piatetsky-Shapiro, G., Smyth, P., Uthurusamy, R., eds. *Advances in knowledge discovery and data mining*: pp. 1 – 34. Menlo Park, CA, USA:AAAI

Freitas, A. A., 1999. A genetic algorithm for generalized rule induction. In Roy, R., Furuhashi, T., Chawdhry, P. K., eds. *Advances in Soft Computing: Engineering Design and Manufacturing*. Berlin:Springer-Verlag, pp. 340 – 353.

Freitas, A. A., 2002. *Data Mining and Knowledge Discovery with Evolutionary Algorithms*. Secaucus, NJ, USA: Springer-Verlag New York, Inc.

Fu, Z., 1999. An innovative GA-based decision tree classifier in large scale data mining. *Principles of Data Mining and Knowledge Discovery*, pp. 348 – 353.

Hand, D. J., Mannila, H., Smyth, P., 2001. *Principles of Data Mining*. The MIT Press

Keijzer, M., Babovic, V., 1999. Dimensionally aware genetic programming. *Proceedings of the Genetic and Evolutionary Computation Conference 1990*, pp. 1069 – 1076. July 13-17, 2005, Orlando, FL, USA.

Koza, J., 1992. *Genetic Programming: On the Programming of Computers by Means of Natural Selection.* Cambridge, MA, USA: The MIT Press.

Langdon, W. B., Poli, R., 2002. *Foundations of Genetic Programming*. Berlin, Germany:Springer-Verlag.

Papagelis, A., Kalles, D., 2001. Breeding decision trees using evolutionary techniques. *Proceedings of the Eighteenth International Conference on Machine Learning (ICML '01)*, pp. 393 – 400. June 28 – July 1 2001, Williamstown, MA, USA.

Rosca, J. P., Ballard, D. H., 1996. Discovery of subroutines in genetic programming. In Angeline, P. J., Kinnear, K. E., eds. *Advances in Genetic Programming 2*, pp. 177 – 202. Cambridge, MA, USA: The MIT Press.

Ryan, M. D., Rayward-Smith, V. J., 1998. The evolution of decision trees. In Koza, J. R., Banzhaf, W., Chellapilla, K., Deb, K., Dorigo, M., Fogel, D. B., Garzon, M. H., Goldberg, D. E., Iba, H., Riolo, R., eds. *Genetic Programming 1998: Proceedings of the Third Annual Conference*, pp. 350 – 358, University of Wisconsin, Wisconsin, USA:Morgan Kaufmann.

Wagner, S., Affenzeller, M., 2005. HeuristicLab: A generic and extensible optimization environment. *Proceedings of the International Conference on Adaptive and Natural Computing Algorithms*, pp. 538 – 541. March 21-23 2005, Coimbra, Portugal.

Winkler, S., Affenzeller, M., Wagner, S., 2007a. Advanced Genetic Programming Based Machine Learning. *Journal of Mathematical Modelling and Algorithms*, Vol. 6: pp. 455 – 480.

Winkler, S., Affenzeller, M., Wagner, S., 2007b. Selection Pressure Driven Sliding Window Genetic Programming. *Computer Aided Systems Theory – EUROCAST 2007, Lecture Notes in Computer Science*, Vol. 4739: pp. 788 – 795.

Wong, M. L., Leung, K. S. 2000. *Data Mining Using Grammar Based Genetic Programming and Applications*, Norwell, MA, USA: Kluwer Academic Publishers

AUTHORS BIOGRAPHY

GABRIEL K. KRONBERGER received his MSc. in computer science in 2005 from Johannes Kepler University Linz, Austria. His research interests include parallel evolutionary algorithms, genetic programming, machine learning and data mining. Currently he is a research associate at the Research Center Hagenberg of the Upper Austrian University of Applied Sciences.

STEPHAN M. WINKLER received his MSc in computer science in 2004 and his PhD in engineering sciences in 2008, both from JKU Linz, Austria. His research interests include genetic programming, nonlinear model identification and machine learning. Currently he is research associate at the Research Center Hagenberg of the Upper Austrian University of Applied Sciences, working on the research program L284-N04 "GP-Based Techniques for the Design of Virtual Sensors", a research project funded by the Austrian Science Fund (FWF).

MICHAEL AFFENZELLER has published several papers and journal articles dealing with theoretical aspects of evolutionary computation and genetic algorithms. In 1997 he received his MSc in mathematics and in 2001 his PhD in engineering sciences, both from JKU Linz, Austria. He is professor at the Upper Austria University of Applied Sciences (Campus Hagenberg) and associate professor at the Institute of Formal Models and Verification at JKU Linz since his habilitation in 2004.

STEFAN WAGNER also received his MSc in computer science in 2004 from Johannes Kepler University Linz, Austria. He currently holds the position of an associate professor at the Upper Austrian University of Applied Sciences (Campus Hagenberg). His research interests include evolutionary computation and heuristic optimization, theory and application of genetic algorithms, machine learning and software development.

OPTIMIZATION OF MEDICAL ULTRASOUND TRANSDUCERS WITH SIMULATION AND GENETIC ALGORITHMS

Monika Kofler(1), Andreas Beham(1), Michael Affenzeller(1), Stefan Wagner(1)

[1] Heuristic and Evolutionary Algorithms Laboratory
Upper Austrian University of Applied Sciences,
School of Informatics, Communications and Media
Softwarepark 11, 4232 Hagenberg, Austria

monika.kofler@heuristiclab.com, andreas.beham@heuristiclab.com,
michael.affenzeller@heuristiclab.com, stefan.wager@heuristiclab.com

ABSTRACT

In recent years, software systems for the simulation of medical ultrasound imaging have been developed, thus making it possible to partially prototype transducers in software. In this paper a set of ultrasound transducer parameters is optimized using a genetic algorithm (GA). The settings are evaluated by simulating a B-mode scan for each transducer configuration. The quality of the generated B-Mode images is assessed by an image quality metric.

Keywords: ultrasound simulation, genetic algorithms, parameter optimization

1. INTRODUCTION

Ultrasound (US) imaging has become an attractive technology for medical diagnosis due to its non-invasive nature, real-time capability and cost effectiveness. Unlike other diagnostic modalities, ultrasound systems do not emit ionizing radiation and scans may be conducted as often as necessary without the risks of repeated exposure to X-rays or radionuclides.

Unfortunately speckle artifacts and low spatial resolution limit the use of ultrasound imaging in clinical practice. A variety of methods have been proposed to improve US image quality and assist physicians in the interpretation of their scans. One possible approach is the utilization of image and signal processing algorithms to remove artifacts and enhance the contrast (cf. Section 2.2). This is mostly a post-processing step, though, taking over after the image has already been captured.

In this study the focus will be on the medical ultrasound scanner instead, and more specifically on the optimization of the transducer probe. The probe is the part of the ultrasound system that emits the sound waves and receives its echoes. It is therefore most crucial for obtaining high-quality US images.

The authors propose a software-based and fully automatic optimization of transducer arrays in order to improve ultrasound image quality. An ultrasound simulator and a genetic algorithm are employed to achieve this goal (cf. Section 4).

2. DIAGNOSTIC ULTRASOUND IMAGING

In medical ultrasound a sound wave is produced by an array of piezoelectric transducers. To ensure good coupling between patient and ultrasound probe a water-based gel is usually applied to the patient's skin before scanning. The transceiver probe emits a pulse of ultrasound and then switches to reception mode to register the reflected echoes. The reflected or backscattered echoes are detected and used to form a digital image of the scanned region.

2.1. Imaging modes

One of the oldest – and most basic – ultrasound modes is the so-called A-mode scan (abbreviation for amplitude scan). A single ultrasound pulse is emitted from the probe. When the pulse crosses the boundary between two tissues of different density, part of the wave is reflected back towards the probe. This echo is detected by the transceiver and displayed as a spike in a line plot.

Brightness mode (also known as B-mode) images are produced by arranging many A-scans parallel to each other and displaying the amplitudes as grey-scale values. The raw radio-frequency (RF) data compromises both amplitude and phase information from the detected waves, the latter of which is discarded during envelope detection. To display the amplitudes the resulting image has to be compressed, as most computer screens only have a dynamic range of 20-30 dB (Dutt 1995).

Other important imaging modes include M-mode and Doppler sonography. For further information on ultrasound technology and instrumentation see for example (Zagzebski 1996).

2.2. Speckle, Noise and Low Resolution

Both the spatial and contrast resolution of ultrasound B-mode scans are limited by speckle artefacts that give the images their characteristic granular structure (Burckhardt 1978). Speckle is caused by tiny particles below the resolution of the ultrasound modality, also called scatterers, which produce a random interference pattern that has no direct link to the biological structure of the scanned tissue.

Since the speckle pattern obscures biological features, blurs tissue boundaries and restricts the contrast resolution of US images, different techniques have been applied to suppress this undesirable effect, among them spatial compounding (Trahey, Smith and von Ramm 1968) and filtering (Loupas, McDicken and Allen 1989, Jin, Silva and Frery 2004).

Insonifying a patch of tissue from different angles, using different ultrasound transducers or pulse length will produce quite different speckle patterns. However, if an object is scanned twice under exactly the same conditions, an identical pattern is obtained (Burckhardt 1978). That is why speckle patterns, although they appear to be random, are not noise in the same sense as for example electrical noise. It would therefore be interesting (in a subsequent stage) to analyze the produced speckle patterns for a number of transducer arrays and identify possible parameter dependencies.

Apart from presenting potential future work, the speckle patterns also strongly influence the choice of the evaluation function in Section 4.4.

3. PREVIOUS WORK

Researchers have used heuristic optimization algorithms, like genetic or evolutionary algorithms, to optimize or estimate ultrasound transducer settings in the past, albeit mostly with a strong focus on electromechanical design parameters.

Fu, Hemsel and Wallaschek (2006) employed evolutionary algorithms to optimize the setup of Langevin ultrasound transducers, also called sandwich transducers. The name is due to their characteristic setup, which consists of (multiple) piezoelectric ceramic disks sandwiched between metallic end blocks. Relevant design parameters include the materials used in the different layers, layer thickness or the number of piezoelectric elements. These transducers are developed for industrial machining, especially high vibration welding and cleaning. The optimization objectives (maximize the vibration amplitude; minimize the electric input power) differ significantly from the objectives for a medical ultrasound transducer (maximize the diagnostic value of generated B-scan images; maximize comfort and safety of the patient), as pursued in this work.

Ruíz, Ramos and San Emeterio (2004) addressed the reverse problem in their work, namely how to estimate the internal design parameters (thickness, acoustic impedance etc.) for an existing ultrasound transducer. This is especially useful for modeling and simulation of ultrasound systems, since the construction details of commercial ultrasound probes are rarely known. In this case the transducer is regarded as blackbox and the internal parameters are estimated from the output signal via a genetic algorithm.

4. A SYSTEM FOR THE OPTIMIZATION OF ULTRASOUND TRANSDUCER SETTINGS

The proposed system consists of the following modules, which will be explained in detail after the general overview of the algorithm:

- **HeuristicLab:** Optimization Framework
- **Field II:** Ultrasound Simulator
- **Image Quality Index** (IQ): Similarity measure between two images

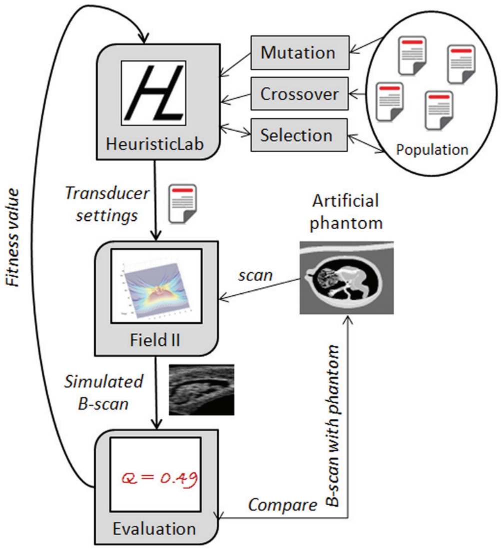


Figure 1: Fully automated optimization cycle

The heuristic optimization environment HeuristicLab provides all the necessary components to build a customized genetic algorithm (Holland, 1992) for the optimization of transducer parameters.

A single transducer is represented by a vector of real values containing parameters such as the center frequency or the number of sensor elements (for a detailed list of the selected parameters see Section 4.2).

As depicted in Figure 1 the optimization cycle works in the following way:

1. Initially HeuristicLab generates a number of vectors (= transducers) which are randomly initialized. In order to generate (preferably better) offspring from this initial population, the quality, also called fitness, of the available transducers has to be assessed first.
2. The parameter vectors are therefore passed on to the ultrasound simulator Field II which simulates a B-mode scan with each transducer and produces an image.
3. This scan is then evaluated by the Image Quality Index module which assigns a fitness value between 0 and 1 to the image (and thus indirectly determines the quality of the transducer).

4. Our GA commences with the usual optimization steps (parent selection, generate new individuals with crossover/mutation, evaluation of new transducers, replacement mechanism etc.).
5. The algorithm terminates after a pre-defined number of iterations or in case of manual abortion of the optimization.

4.1. HeuristicLab

HeuristicLab (Wagner et al. 2007) is a paradigm-independent and highly customizable optimization framework. HeuristicLab provides a number of standard workbenches for frequently used algorithms, but its plug-in based architecture allows users to integrate their own problem representations or domain-specific operators into the system with ease.

4.2. Simulation with Field II

The ultrasound simulation program Field II (Jensen and Svendsen 1992, Jensen 1996) by Jørgen Ardendt Jensen from the Technical University of Denmark was used as simulator in this study. Field II relies on linear system theory to model the ultrasound field emitted from the transducer. In ultrasound imaging the impulse response varies in relation to the position relative to the transducer, hence the name spatial impulse response. Field II can approximate the spatial impulse response for arbitrary transducer settings and geometries.

4.3. Parameters for optimization

Field II simulates ultrasound transducers of arbitrary shape and facilitates an assortment of custom parameter settings, including dynamic focusing and apodization (= reduced vibration of the transducer surface with distance from its center), different excitation pulses, concave and even sparsely populated transducer arrays (Jensen 1996).

For this study the parameters subject to the optimization were limited to a selected few, namely:

- **f0**: Transducer center frequency
  Range: 2.5 – 15MHz
  A higher frequency generates a focused ultrasound beam that yields better image resolution at the cost of reduced imaging depth.
- **fs:** Sampling frequency
  Range: 0–200 MHz
- **element_height:** Height of one element
  Range: 0-10 mm
- **element_width:** Width of one element
  Range: 0.1-1 mm
- **kerf:** Spacing between elements in lateral direction
  Range: 0.1-5 mm
- **focus:** Fixed focal point (*lateral, elevational, axial*)
  At the focal point the sound field constructively converges. The focal point is given relative to the center of the transducer surface which is defined as (0,0,0). The axial location, also called the focal depth, is of particular interest.
  Range (for axial location): 0-100 mm
- **n_elements:** Number of physical elements in lateral direction (= one-dimensional array)
  Possible Values: 2, 4, 8, 16, 32, 64, 128, 256, 512 and 1024

Figure 2 shows the physical dimensions of a transducer and its main parameters (except for the transducer frequency). Each setting of the parameter vector *(f0, fs, element_height, element_width, kerf, focus, n_elements)* describes an individual ultrasound transducer and serves as representation of a solution candidate for the genetic algorithm The quality of each GA-generated solution candidate is assessed by simulation of the transducer with Field II and subsequent evaluation of the generated ultrasound B-mode image (cf. Section 4.5).

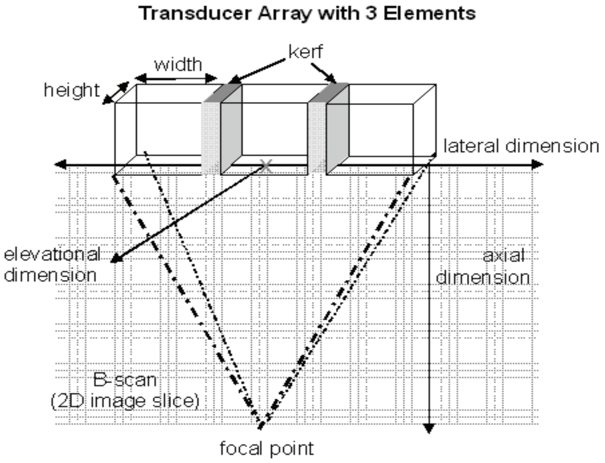


Figure 2: Transducer parameters and nomenclature for the ultrasound imaging directions (axial, lateral, elevational).

4.4. Simulating a B-mode scan

To assess the quality of ultrasound transducers a variety of clinical requirements has been identified by Angelsen et al. (1995), among them spatial resolution, imaging artefacts and sensitivity. In addition, anatomical constraints, as encountered in the fields of cardiac or pelvic ultrasound, and patient comfort may pose restrictions on both the size and shape of the transducer probe. For example, to scan the chambers and valves of the heart the ultrasound beam has to be aimed through a narrow opening between the ribs, since both bone and lung tissue obstruct the passage of ultrasound waves. Transthoracic transducer probes are therefore usually narrow and long. Similar restrictions apply for gynecological exams, breast ultrasound screenings or transrectal exams.

Such application specific considerations aside, a medical ultrasound system is mostly judged by the quality – in terms of diagnostic value – of the ultrasound images it produces. Therefore the authors decided not use physical target parameters (like the vibration amplitude) to evaluate transducers but to perform an examination on a "virtual patient" and rate the transducers by the image quality they produce.

A synthetic phantom of a fetus in the third month of development (Jensen and Munk 1997) was used for the evaluation of the transducer setting (cf. Figure 3). For each generated transducer array a scan of the phantom was simulated with Field II. One image scan consists of 128 raw radio-frequency (RF) lines that can be calculated in parallel to speed up the simulation. The generated RF-data was then post-processed for display on a computer screen by discarding the phase information and performing logarithmic compression, thus yielding a simulated B-mode scan of the phantom.

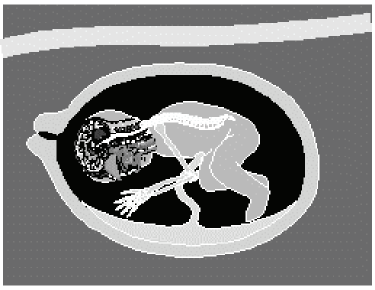


Figure 3: Synthetic phantom of a fetus in the third month of development (Jensen and Munk 1997): The varying grayscale values depict the scattering strength and density of the modeled tissues. The phantom is generated with 200.000 randomly distributed scatterers.

The evaluation of B-mode images is usually performed by human experts. While this is certainly the most accurate and important method of assessing image quality for medical diagnosis, the generated images were rated by employing the image quality index (IQ) proposed by Wang and Bovik (2002) to create a fully automatic optimization cycle as depicted in Figure 1.

4.5. Image quality index

The grainy appearance of ultrasound B-mode images is not only a diagnostic issue but also makes image processing very difficult. As reported by (Pluim, Maintz and Viergever 2003) it is often necessary to smooth the ultrasound images (e.g. by low-pass filtering) before common medical image processing algorithms for feature detection or registration can be applied.

The image quality index was selected for fitness evaluation because it works well under different kinds of image distortions, including multiplicative speckle noise (Wang and Bovik 2002). The method takes the phantom image and the generated image and calculates a similarity measure scaled between 0 and 1.

5. RESULTS AND DISCUSSION

A standard GA with a population size of 10 individuals was used to test the optimization concept. The initial population was generated with a random parameter setting. Due to the long simulation times, especially for transducers with large *n_elements*, the algorithm completed three iterations and produced a total of 43 transducer settings that were evaluated so far. Two aspects have affected the optimization test run severely, namely

- simulation time requirements and
- issues with the image quality index IQ

which shall be elaborated in more detail in the following sections.

5.1. Simulation time requirements

Simulating a single RF line took 25 minutes on average on a Pentium 4 CPU, 3GHz with 1GB RAM. Since one generated B-mode image consists of 128 such lines, simulations were conducted in parallel on 10 Pentium 4 computers to reduce the total processing time.

Despite the parallelization of the ultrasound simulation, evaluating a single population of transducer parameter settings can take a couple of days, thus the number of individuals per generation was limited to 10. The processing time per RF line ranged between 20 seconds and 14 hours, depending on the individual parameter settings. Experiments with 1024 *n_elements* were tried as well but proved to have a very unfavorable simulation time (approximately 1 week per B-mode image, with parallel simulation on 10 workstations).

The authors expect that transducer setting with larger *n_elements* will dominate the optimization as the number of iterations increases, thereby requiring more and more processing time for the evaluation. The currently available parallel infrastructure with 10 desktop computers, which can be utilized for nightly evaluation runs, is not sufficiently dimensioned to complete such a task within an acceptable time span. An anticipated expansion of our simulation cluster will improve the scaling of the algorithm for transducers with large *n_elements*, thus alleviating the issue of increasing simulation times.

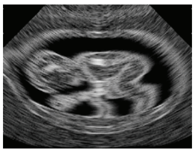
Alternatively, since the RF lines can be simulated independently of each other, one could define a region of interest (ROI) of perhaps 20 lines which are simulated first in case of a large *n_elements* value. Subsequently the fitness of the ROI is calculated and used to assess whether the simulation of the whole image would pay off or if that particular parameter set should be discarded immediately.

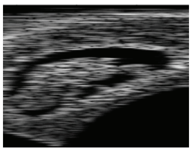
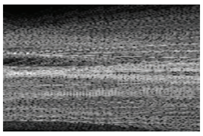
5.2. Fitness evaluation

To get a better appraisal for the fitness values of the generated transducers a reference fitness value from a transducer that was known to be feasible was needed.
The authors used Jensen and Munk's (1997) original parameter settings from the paper to generate the B-mode image that is displayed in Table 1 as #1. The fitness value of 0.552 shows the impact the presence of speckle has on image-similarity, even for parameter settings that are surely very good.

As described in Section 4.5 the image quality index IQ does indeed work well despite the presence of speckle patterns in the generated images. What the authors had not anticipated were the spatial deformations of the target B-scan (see Table 1, #2) that were produced by some of the settings. Since the image quality index was not designed to do registration of medical US images as well, this led to IQ rankings that did not sustain a visual inspection, as in the case of scan #2 and #3 in Table 1. A further refinement of the fitness evaluation is therefore defenitely adviseable.

Table 1: Parameter settings for selected transducers, their respective simulated B-scans and fitness values. The elements of the parameter vector and their units are (f0[MHz], fs[MHz], element_height[mm], element_width[mm], kerf[mm], focus[mm], n_elements)

#1

	Reference Configuration (Jensen and Munk 1997):
	Settings:
	`(5, 100,7, 0.308, 2.5, 70, 64)`
	F = 0.552

#2 	Good solution from GA Settings: `(7.379, 129.005, 4.81, 1.263, 0.199, 81.31, 64)` **F = 0.432**
#3 	Bad solution from GA Settings: `(4.327, 155.42, 0.933, 6.187, 0.5327, 40.679, 16)` **F = 0.49**
#4 	Worst simulated solution Settings: `(12.77, 12.16, 0.558, 4.529, 0.667, 90.993, 256)` **F = 0.0827**

6. CONCLUSION AND FUTURE PERSPECTIVES

The authors showed that it is possible to optimize medical ultrasound transducer arrays autonomously using a genetic algorithm, an ultrasound simulation program and an image quality metric. The main issues of the approach in its current form are the duration of the simulation runs and an image quality metric that is still far from optimal.

Nevertheless, the authors are convinced that the optimization of ultrasound transducers via simulation and genetic algorithms might be an attractive option for ultrasound manufacturers to reduce their hardware prototyping efforts. Since the algorithm works autonomously, it could schedule the RF-line calculations to run on unoccupied desktop computers during the night and/or on weekends, and calculate a pre-selection of promising transducer geometries for subsequent hardware prototyping and clinical tests.

By extending the parameter set, e.g. by allowing sparsely populated arrays, even new transducer geometries might be found – always assuming that the simulation yields valid results for unusual settings. The authors plan to consult ultrasound experts in the next development step to discuss the simulation results and to further refine both the image quality index and the parameter set for optimization.

REFERENCES

Angelsen, B.A.J., Torp, H., Holm, S., Kristoffersen, K. and Whittingham, T.A., 1995. Which transducer array is best? *European Journal of Ultrasound*, Volume 2, Number 2, pp. 151-164.

Burckhardt, C.B., 1978. Speckle in ultrasound b-mode scans. *IEEE Transactions on Sonics and Ultrasonics*, Volume 25, Issue 1, pp. 1-6.

Dutt, V., 1995. *Statistical Analysis of Ultrasound Echo Envelope.* PhD thesis, The Mayo Graduate School

Fu, B., Hemsel, T. and Wallaschek, J., 2006. Piezoelectric transducer design via multiobjective optimization. *Ultrasonics*, Volume 44, Supplement 1, pp. e747-e752.

Holland, J.H., 1992. *Adaptation in Natural and Artificial Systems: 2nd edition.* MIT Press.

Jensen, J.A., 1996. Field: A Program for Simulating Ultrasound Systems, Paper presented at the *10th Nordic-Baltic Conference on Biomedical Imaging* Published in *Medical & Biological Engineering & Computing*, Volume 34, Supplement 1, Part 1, pp. 351-353.

Jensen, J.A. and Svendsen, N. B., 1992. Calculation of pressure fields from arbitrarily shaped, apodized, and excited ultrasound transducers, *IEEE Transactions on Ultrasonics, Ferroelectrics and Frequency Control*, Volume 39, pp. 262-267.

Jensen, J.A. and Munk, P., 1997. Computer phantoms for simulating ultrasound B-mode and cfm images, *23rd Acoustical Imaging Symposium*, April 13-16, Boston, Massachusetts, USA.

Jin, J.Y, Silva, G.T. and Frery, A.C., 2004. SAR despeckling filters in ultrasound imaging. *Latin American Applied Research*, Volume 34, Issue 1, pp. 49-53.

Loupas, T., McDicken, W.N. and Allen, P.L., 1989. Adaptive weighted median filter for speckle suppression in medical ultrasonic images. *IEEE Transactions on Circuits and Systems*, Volume 36, pp. 255-269.

Pluim, J.P.W., Maintz, J.B.A. and Viergever, M.A., 2003. Mutal-information-based registration of medical images: a survey. *IEEE Transactions on Medical Imaging*, Volume 22, pp. 986-1004.

Ruíz, A., Ramos, A. and San Emeterio, J.L., 2004. Estimation of some transducer parameters in a broadband piezoelectric transmitter by using an artificial intelligence technique, *Ultrasonics*, Volume 42, pp. 459-463.

Trahey, G.E., Smith, S.W. and von Ramm, O.T., 1968. Speckle pattern correlation with lateral aperture translation: experimental results and implications for spatial compounding. *IEEE Transactions on Ultrasonics, Ferroelectrics and Frequency Control,* Volume 33, Issue 3, pp. 257-264.

Wagner, S., Winkler, S., Braune, R., Kronberger, G., Beham, A. and Affenzeller, M., 2007. Benefits of Plugin-Based Heuristic Optimization Software Systems, *Computer Aided Systems Theory - EUROCAST 2007*, Lecture Notes in Computer Science 4739, pp. 747-754. Springer-Verlag.

Wang, Z. and Bovik, A.C., 2002. A universal quality index, *IEEE Signal Processing Letters*, Volume 9, Number 3, pp. 81-84.

Zagzebski, J.A., 1996. *Essentials of ultrasound physics*. Mosby

AUTHORS BIOGRAPHY

MONIKA KOFLER studied Medical Software Engineering at the Upper Austrian University of Applied Sciences, Campus Hagenberg, Austria, from which she received her diploma's degree in 2006. She is currently employed as a research associate at the Research Center Hagenberg and pursues her PhD in engineering sciences at the Johannes Kepler University Linz, Austria.

ANDREAS BEHAM received his MSc in computer science in 2007 from the Johannes Kepler University (JKU) Linz, Austria. His research interests include heuristic optimization methods, simulation and combinatorial optimization. Currently he is a PhD student at the JKU and a research associate at the Research Center Hagenberg of the Upper Austria University of Applied Sciences (Campus Hagenberg).

MICHAEL AFFENZELLER has published several papers and journal articles dealing with theoretical aspects of genetic algorithms and evolutionary computation in general. In 1997 he received his MSc in Industrial Mathematics and in 2001 his PhD in Computer Science, both from the Johannes Kepler University Linz, Austria. He is professor at the Upper Austria University of Applied Sciences (Campus Hagenberg) and associate professor at the Institute of Formal Models and Verification at Johannes Kepler University Linz, Austria since his habilitation in 2004.

STEFAN WAGNER received his MSc in computer science in 2004 from Johannes Kepler University Linz, Austria. He now holds the position of an associate professor at the Upper Austria University of Applied Sciences (Campus Hagenberg). His research interests include evolutionary computation and heuristic optimization, theory and application of genetic algorithms, machine learning and software development.

The web pages of all members of the *Heuristic and Evolutionary Algorithms Laboratory* as well as further information about HeuristicLab and related scientific work of the can be found at:
http://www.heuristiclab.com.

MODELING OF HEURISTIC OPTIMIZATION ALGORITHMS

Stefan Wagner[(a)], Gabriel Kronberger[(b)], Andreas Beham[(c)], Stephan Winkler[(d)], Michael Affenzeller[(e)]

[(a), (b), (c), (d), (e)]Heuristic and Evolutionary Algorithms Laboratory
School of Informatics, Communications and Media – Hagenberg
Upper Austria University of Applied Sciences
Softwarepark 11, A-4232 Hagenberg, Austria

[(a)]stefan.wagner@heuristiclab.com, [(b)]gabriel.kronberger@heuristiclab.com, [(c)]andreas.beham@heuristiclab.com,
[(d)]stephan.winkler@heuristiclab.com, [(e)]michael.affenzeller@heuristiclab.com

ABSTRACT

The definition of a generic algorithm model for representing arbitrary heuristic optimization algorithms is one of the most challenging tasks when developing heuristic optimization software systems. As a high degree of flexibility and a large amount of reusable code are requirements that are hard to fulfill together, existing frameworks often lack of either of them to a certain extent. To overcome these difficulties the authors present a generic algorithm model not only capable of representing heuristic optimization but that can be used for modeling arbitrary algorithms. This model can be used as a meta-model for heuristic optimization algorithms, enabling users to represent custom algorithms in a flexible way by still providing a broad spectrum of reusable algorithm building blocks.

Keywords: heuristic optimization, algorithm modeling, software engineering, software tools

1. INTRODUCTION

In the last decades a steady increase of computational resources and concurrently an impressive drop of hardware prices could be observed. Nowadays, very powerful computer systems are found in almost every company or research institution revealing a processing power one could only dream of a few years ago. This trend opens the door for attacking complex optimization problems from various domains that were not solvable in the past. Concerning problem solving methodologies especially heuristic algorithms are very successful in that sense, as they provide a reasonable compromise between solution quality and required runtime.

In the research area of heuristic algorithms a broad spectrum of optimization techniques has been developed. In addition to problem-specific heuristics, particularly the development of meta-heuristics is a very active field of research as these algorithms represent generic methods that can be used for solving many different optimization problems. Thereby a huge variety of often nature inspired archetypes has been used as a basis for developing new optimization paradigms like evolutionary algorithms, ant systems, particle swarm optimization, tabu search, or simulated annealing. Several publications show successful applications of such meta-heuristics in various problem domains. A recent overview is for example given in (Doerner, Gendreau, Greistorfer, Gutjahr, Hartl, and Reimann 2007).

Today, this broad spectrum of different algorithmic concepts makes it more and more difficult for researchers to compare new algorithms with existing ones to show advantageous properties of some new approach. As most research puts a focus on one particular heuristic optimization paradigm, comparisons with other algorithms are often not quite fair and objective. In many cases a thoroughly optimized algorithm containing cutting-edge concepts is compared with standard and non-optimized algorithms of other paradigms representing research know-how that is several years old.

One of the reasons for these difficulties is that there is no common model for heuristic optimization algorithms in general that can be used to represent, execute and compare arbitrary algorithms. Many existing software frameworks focus on one or a few particular optimization paradigms and miss the goal of providing an infrastructure generic enough to represent all different kinds of algorithms.

In this paper the authors try to overcome this problem by shifting the layer of abstraction one level up. Instead of trying to incorporate different heuristic optimization algorithms into a common model, a generic algorithm (meta-)model is presented that is capable of representing not only heuristic optimization but arbitrary algorithms. By this means the model can be used for developing custom algorithm models for various optimization paradigms. Furthermore, by considering aspects like parallelism or user interaction on different layers of abstraction the presented algorithm model can serve as a basis for development of a next generation heuristic optimization environment that can be used by many researchers to rapidly develop and fairly compare their algorithms.

2. EXISTING SOFTWARE SYSTEMS FOR HEURISTIC OPTIMIZATION

Today, modern concepts of software engineering like object-oriented or component-oriented programming represent the state of the art for creating complex software systems by providing a high level of code reuse, good maintainability and a high degree of flexibility and extensibility (Johnson and Foote 1988). However, such approaches are not yet established on a broad basis in the area of heuristic optimization, as this field is much younger than classical domains of software systems (like word processing, spread sheets, image processing, or integrated development environments). Most systems for heuristic optimization are one man projects and are developed by researchers or students to realize one or a few algorithms for attacking a specific problem. Naturally, when a software system is developed mainly for personal use or a very small, well known and personally connected user group, software quality aspects like reusability, flexibility, genericity, documentation and a clean design are not the primer concern of developers. As a consequence, these applications still suffer from a quite low level of maturity seen from a software engineering point of view.

In the last years and with the ongoing success of heuristic algorithms also in commercial areas, the heuristic optimization community started to be aware of this situation. Advantages of well designed, powerful, flexible and ready-to-use heuristic optimization frameworks were identified and discussed in several publications like (Voß and Woodruff 2002; Jones, McKeown, and Rayward-Smith 2002; Gagne and Parizeau 2006). Furthermore, some research groups started to head for these goals and began redesigning existing or developing new heuristic optimization software systems which were promoted as flexible and powerful black or white box frameworks available and useable for a broad group of users in the scientific as well as in the commercial domain. In comparison to the systems available before, main advantages of these frameworks are on the one hand a wide range of ready-to-use classical algorithms, solution representations, manipulation operators and benchmark problems which make it easy to jump into the area of heuristic optimization and to start experimenting and comparing various concepts. On the other hand a high degree of flexibility due to a clean object-oriented design makes it easy for users to implement custom extensions like specific optimization problems or algorithmic ideas.

One of the most challenging tasks in the development of such a general purpose heuristic optimization framework is the definition of an object model representing arbitrary heuristic optimization paradigms. On the one hand this model has to be flexible and extensible to a very high degree so that users can integrate non-standard algorithms that often do not fit into existing paradigms exactly. On the other hand this model should be very fine granular so that a broad spectrum of existing classical algorithms can be represented in form of algorithm modules. These modules can then serve as building blocks to realize different algorithm variations or completely new algorithms with a high amount of reusable code.

One main question is on which level of abstraction such a model should be defined. A high level of abstraction leads to large building blocks and a very flexible system. A lower level of abstraction supports reusability by providing many small building blocks, but the structure of algorithms has to be predefined more strictly in that case which reduces flexibility.

Taking a look at several existing frameworks for heuristic optimization, it can be seen that this question has been answered in quite different ways. For example, in the Templar framework developed by Martin Jones and his colleagues at the University of East Anglia (Jones, McKeown, and Rayward-Smith 2002) a very high level of abstraction has been realized. In Templar each algorithm is represented as so-called engines. Although, the framework supports distribution, hybridization and cooperation of engines, no more fine granular representation of algorithms is considered. Therefore, when a new algorithm with just a slight modification of an existing one is required, the engine of the existing algorithm has to be copied and modified leading to code duplication and less maintainability.

As another example the HotFrame framework developed by Andreas Fink and his colleagues at the University of Hamburg (Fink and Voß 2002) provides a very low level of abstraction. HotFrame contains a large amount of generic C++ classes that can be put together to represent an algorithm. However, in that way the basic algorithm model is more strictly predefined forcing users to fit their custom algorithms into that class structure. Furthermore, due to the complexity of the model the framework also suffers from a quite steep learning curve.

Obviously neither a high nor a low level of abstraction is able to fulfill both, a high degree of flexibility and reusability of code, as these two requirements can be considered as mutually exclusive.

3. GENERIC ALGORITHM MODEL

In order to overcome this problem the authors decided to use a different approach. Instead of trying to develop an algorithm model representing all different kinds of heuristic optimization algorithms, a generic algorithm (meta-)model inspired by classical programming languages is presented in this paper. This model is generic enough to represent not only heuristic optimization techniques but all different kinds of algorithms in general.

On top of this generic algorithm model more specific models for representing heuristic optimization algorithms can be defined. These specific models do not have to be hard-coded in a framework though, but can be defined on the user level. A large variety of different algorithm models can be realized, opening the door for each user to either reuse an existing one or to create an own model if necessary. By shifting the model one layer up, users do not need to fit custom algorithms into a single fixed model but can fit the model itself to their needs in order to be able to represent their algorithms.

From an abstract point of view an algorithm is a sequence of *steps* (operations, instructions, statements) describing manipulation of *data* (variables) that is finally executed by a *machine* (or human). Consequently, these three aspects (data, operators and execution) represent the core components that have to be represented by the model and are considered in the following sections.

3.1. Data Model

In classical programming languages variables are used to represent data values manipulated in an algorithm. Variables link a data value with a (human readable) name and (optionally) a data type so that they can be referenced in the statements and instructions manipulating the data. This concept is also taken up in the data model. A variable object is a simple key-value-pair containing a name (represented as a string) and a value (an arbitrary object). The data type of a variable's value doesn't have to be fixed explicitly but is given by the type of the value itself.

In a typical heuristic optimization algorithm a lot of different data values and consequently also variables are used. Hence, in addition to data values and variables special objects called scopes are needed for variable management to keep a clear structure. Each scope can hold an arbitrary number of variables. To access a variable in a scope the variable name is used as an identifier, so each variable has to have a unique name in each scope it is contained.

In the domain of heuristic optimization hierarchical structures are very common. For example, in terms of evolutionary computation, an environment contains several populations, each population contains individuals (solutions) and these solutions may consist of different solution parts. Furthermore, hierarchical structures are not only very suitable in the area of heuristic optimization but in general are used to assemble complex data structures by combining simple ones. As a consequence it is quite reasonable to combine scopes in a hierarchical way to represent such layers of abstraction. Each scope may contain any number of sub-scopes leading to an n-ary tree structure. For example one scope representing a set of solutions (population) may contain several other (sub-)scopes representing the solutions themselves.

When retrieving a variable from a scope this hierarchical structure of scopes is also taken into account. If a variable (identified by its name) is not found in a scope, the variable lookup mechanism continues searching for the variable in the parent scope of the current scope. The lookup is continued as long as the variable is not found and as long as there is another parent scope left (i.e. until the root scope is reached). Consequently, each variable in a scope is also "visible" in all sub-scopes of that scope. However, if another variable with the same name is added in one of the sub-scopes, it hides the original one (due to the lookup procedure). Note that this behavior is very similar to scopes in classical programming languages. That is also the reason why the name "scope" was chosen.

Based on this abstract representation of data, the next section describes operators which are applied on scopes to manipulate data. Therefore, operators represent the fundamental building blocks of algorithms. Due to the hierarchical nature of scopes operators may be applied on different abstraction levels leading to several essential benefits concerning parallelization discussed later on.

3.2. Operator Model

Regarding to the definition of an algorithm, the next topic to be defined are steps. Each algorithm is a sequence of clearly defined, unambiguous and executable instructions. These atomic building blocks of algorithms are called operators and are of course also considered as objects in a generic algorithm model. In analogy to classical programming languages these operators can be seen as statements that represent instructions or procedure calls.

In general, operators fulfill two major tasks: On the one hand an operator can access and manipulate a scope's variables or sub-scopes and on the other hand an operator may define the further execution flow (i.e. which operators are executed next). To support genericity of operators and to enable reuse, operators have to be decoupled from concrete variables. For that reason a mechanism is used that is similar to procedure calls.

As an example consider a simple increment operator that increases the value of an integer variable by one. Inside the operator it is defined that the operator is expecting a variable of a specific type (in our case an integer) and how this variable is going to be used. When implementing an operator, formal names are used to identify variables but these formal names do not correspond to any real variable name. The concrete variable remains unknown until the operator is applied on a scope. By this means an increment operator can be used to increment any arbitrary integer variable. When adding an operator to an algorithm the user has to define a mapping between the formal variable names used inside the operator and the real variable names that should be used when the operator is finally executed. When a variable is accessed by the operator the variable's formal name is automatically translated into the actual name, which is then used to retrieve the variable from the scope. As a consequence meta-information has to be provided by an operator to declare on which variables the operator is going to work on. This information is represented by objects called variable infos that can be added to each operator. Additionally, the user can access these variable infos to set the actual variable names. In analogy to classical procedure calls variable infos can therefore be interpreted as parameter lists of operators.

In order to build complex algorithms, operators are combined to a sequence of operations. Each operator contains arbitrary many references to other operators (sub-operators) representing the static structure of an algorithm. When an operator is executed it can decide which operators have to be executed next. In that way designated control operators can be built that do not manipulate data but dynamically define the execution flow. For example, a sequence of operators can be specified using an operator that just returns all its sub-operators as the next operators to be executed. Another example would be a branch operator that is choosing one of its sub-operators as the next operator depending on the value of some variable contained in the scope the operator is applied on (cf. an if- or switch-statement in classical programming languages). In contrast to scopes, operators do not form a hierarchical structure (although contained operators are called sub-operators) but are combined in a graph. In other words an operator that has already been used in some upper level can be added as a sub-operator again leading to cycles in operator references. In combination with sequences and branches this concept can be easily used to build loops or any other form of control structures known from classical programming languages. For example, a do-while-loop can be realized as a sequence operator containing a branch operator as its last sub-operator. This branch operator can contain a reference back to the sequence operator as its sub-operator defining the branch executed if the condition holds.

As a result, it is possible to represent concepts known from classical (procedural) programming languages in the operator model (sequences, branches, loops). It is therefore capable of representing arbitrary algorithms and of course especially heuristic optimization algorithms.

3.3. Execution Model

The last aspect to be considered is execution of algorithms. Represented as operator graphs algorithms are executed step by step by virtual machines called engines. In each iteration an engine performs an operation which is applying an operator on a scope. Therefore, an operation represents a tuple of an operator and the scope the operator should be applied on. At the beginning of each algorithm execution an engine is initialized with a single operation containing the initial (root) operator of the algorithm and an empty scope (i.e. the global scope).

As the program flow is dynamically defined by operators themselves, each operator may return one or more operations after its execution that have to be executed next. As a consequence engines have to keep track of all operations waiting for execution. These pending operations are kept in a stack. In each iteration an engines pops the next operation from the top of its stack, executes the operator on the scope and pushes the returned successor operations in reverse order back on the stack again (reversing the order is necessary to maintain the execution sequence as a stack is a last-in-first-out queue). By this means engines perform a depth-first expansion of operators. A pseudo-code representation of the main loop of engines is shown below:

```
clear global scope
clear operations stack
push initial operation

WHILE NOT operations stack is empty DO BEGIN
  pop next operation
  apply operator on scope
  push successor operations
END WHILE
```

As a summary of the generic algorithm model consisting of the three parts described in the previous sections (data model, operator model, execution model), figure 1 gives the main identified components and shows the corresponding interactions.

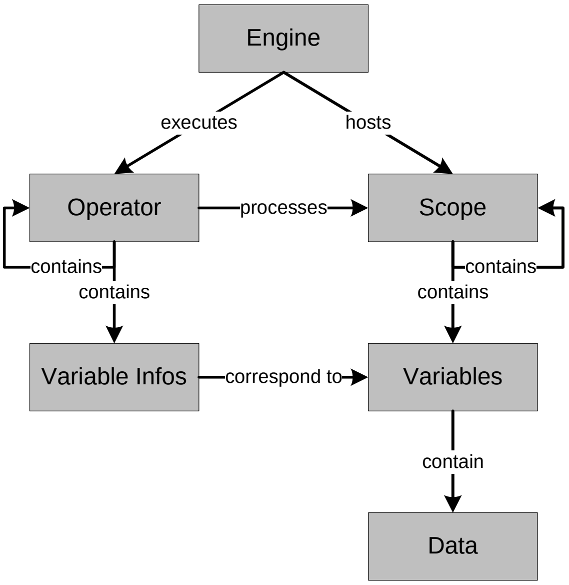


Figure 1: Generic algorithm model

4. PARALLELISM

In many real world applications of heuristic optimization performance is of crucial importance. Therefore, concepts of parallel and distributed computing have to be used frequently to utilize multiple cores or even computers (clusters) and to distribute the work load. In the parallel heuristic optimization community several models of parallelization have been developed reflecting different strategies. In general these models can be categorized in two main approaches:

On the one hand quality calculation can be considered for parallelization. For many optimization problems the computational effort required for calculating the quality of a single solution is much higher than the effort needed by solution manipulation operations. Consider for example heuristic optimization in the area of production planning or logistics. In that case evaluating a solution is done by building a schedule of all jobs or vehicles available whereas manipulation of solutions is usually reduced to twisting of permutations (this depends on the solution encoding but variations of permutation-based encoding are frequently used for combinatorial optimization problems and have been successful in many applications). As another example heuristic optimization of data representation or simulation models can be mentioned as in these applications solution evaluation means executing the whole model (i.e. performing the simulation or checking the quality of the model for all training data). In both examples (and there are many more) executing the evaluation of solutions in parallel is a helpful approach (usually called *global parallelization*) (Alba 2005). However, the heuristic algorithm performing the optimization by creating new and hopefully better solutions remains a sequential one.

On the other hand parallelization can also be considered for heuristic optimization algorithms directly (Alba 2005). By splitting solution candidates into distinct sets, these sets can be optimized independently from each other and therefore in parallel. For example parallel multi-start heuristics are simple representatives of that concept. In that case multiple optimization runs are executed with different initial solutions to achieve a broader coverage of the search space. No information is exchanged between solution sets until the end of the optimization. In more complex approaches exchange of information from time to time is additionally used to keep the search process alive and to support diversification of the search (coarse- or fine-grained parallel genetic algorithms, e.g.). In general, population-based heuristic optimization algorithms are very well suited for this kind of parallelization as multiple populations can be used as distinct sets and no additional splitting of solutions is necessary.

By separating the definition of parallelism in algorithms from the concrete way how a parallel algorithm is executed, users of heuristic optimization software systems can focus on algorithm development without having to rack their brains on how parallelization is actually done. If the basic algorithm model already supports parallelism, all different kinds of parallel algorithms can be modeled enabling also the implementation of these different parallelization strategies used in heuristic optimization discussed above.

The generic algorithm model discussed in this paper follows a strict separation of data, operations and algorithm execution. As a consequence introducing parallelism can be done quite easily by grouping operations into sets that are allowed to be executed in parallel. As an operator may return several operations to be executed next, it can mark this group of successor operations as a parallel group. This signals the engine that some operations are independent from each other and the engine is now free to decide which kind of parallel processing should be used for their execution. How parallelization is actually done depends on the engine only. For example, one engine can be developed that doesn't care about parallelism at all and executes an algorithm still in a sequential way (which is especially helpful for testing algorithms before they are really executed in parallel). Another engine might use multiple threads to execute operations of a parallel group (exploiting multi-core CPUs) or an even more sophisticated engine might distribute parallel operations to several nodes in a network following either a client-server-based or a peer-to-peer based approach (utilizing cluster or grid systems). Also meta-engines are possible that use other engines for execution which enables hybrid parallelization on different levels (for example distributing operations to different cluster nodes on a higher level and using shared-memory parallelization on each node on a lower level). As a consequence the parallelization concept used for executing parallel algorithms can simply be specified by the user by choosing an appropriate engine the algorithm is executed on. The algorithm itself doesn't have to be modified at all.

Based on this parallelization concept the generic algorithm model allows development of special control operators for parallel algorithms. For example, parallel execution of operators can be realized by an operator very similar to the sequence operator already discussed in the previous section. The only difference is that in the parallel case the operator has to mark its successor operations containing all its sub-operators and the actual scope as a parallel group. Furthermore, the hierarchical structure of scopes enables data partitioning in a very intuitive way. As an example consider a sub-scopes processor which returns a parallel group of operations, containing an operation for each of its sub-operators being executed on one of the sub-scopes of the current scope. By this means parallelization can be applied on any level of scopes leading to global, fine- or coarse-grained parallel heuristic algorithms.

5. LAYERS OF USER INTERACTION

As the generic algorithm model described in the previous sections is not dedicated to heuristic optimization but can represent arbitrary algorithms, it offers a very high degree of flexibility. Operators representing a broad spectrum of actions ranging from trivial increments or variable assignments to complex selection or manipulation techniques can be used as building blocks for algorithm development leading to a high degree of code reuse. Furthermore, also custom operators can be integrated easily, if the set of predefined operators provided by a framework is not sufficient.

However, such a low level of abstraction is not reasonable for many users as even the representation of simple algorithms results in quite large and complex operator graphs. Therefore, several layers of user interaction are required that represent different degrees of abstraction.

Such layers can be realized on top of the generic algorithm model by using combined operators that encapsulate operator graphs (i.e. algorithms) and represent more complex operations. In that case the generic algorithm model serves as a meta-model for heuristic optimization algorithms. An important aspect is that combined operators are not hard-coded in a framework but can be developed and shared on the user instead of the development level.

For example, users can provide combined operators representing ready-to-use heuristic optimization algorithms (like a canonical genetic algorithm, simulated annealing, hill climbing, tabu search, or particle swarm optimization) that can be used as black box solvers. By this means other users can start working with specific algorithms right away without having to worry about how an algorithm is structured in detail.

In between predefined solvers and the generic algorithm model, arbitrary other layers can be realized representing various (user-specific) heuristic optimization models. For example, generic models of specific heuristic optimization algorithm flavors (evolutionary algorithms, local search algorithms, etc.) can be represented by a set of operators useful to enable experimenting with these paradigms without putting the burden of the whole complexity and genericity of the basic algorithm model on users.

In figure 2 this layered structure of user interaction is shown schematically. As all these layers use the same algorithm model as their basis, users are free to decide which level of abstraction is adequate for their needs.

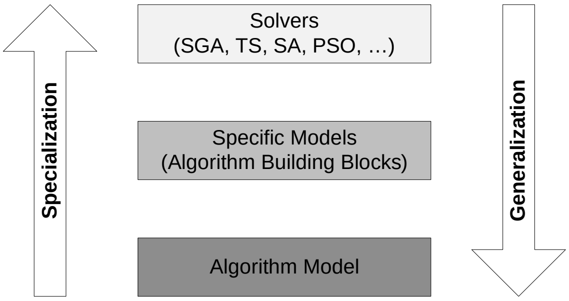


Figure 2: Layers of iser interaction

6. CONCLUSION

In this paper the authors presented a generic model for heuristic optimization algorithms. Compared to other models known from existing heuristic optimization software frameworks, the main advantage of the proposed solution is a higher level of abstraction. By considering the three main aspects of algorithms in general (data, operators and execution), a generic model was developed that is not only capable of representing heuristic optimization techniques but can be used for modeling arbitrary algorithms.

By this means the model can act as a meta-model which enables users to incorporate custom heuristic optimization paradigms and algorithms in a flexible way. As the burden of a single and fixed representation trying to cover all different kinds of heuristic optimization concepts is removed, users are free to realize custom models built on top of the generic algorithm model that exactly fit their needs.

Furthermore, aspects like parallelism or user interaction on different layers of abstraction have been considered, showing that the described model is suitable for developing a new generation of heuristic optimization software systems.

REFERENCES

Alba, E., 2005. *Parallel Metaheuristics: A New Class of Algorithms*. Wiley.

Doerner, K. F., Gendreau, M., Greistorfer, P., Gutjahr, W., Hartl, R. F., Reimann, M., 2007. *Metaheuristics: Progress in Complex Systems Optimization*. Springer.

Fink, A., Voß, S., 2002. HotFrame: A Heuristic Optimization Framework. In: *Optimization Software Class Libraries*. Kluwer.

Gagne, C., Parizeau, M., 2006. Genericity in Evolutionary Computation Software Tools: Principles and Case-Study. *International Journal on Artificial Intelligence Tools*, 15, 173-194.

Johnson, R., Foote, B., 1988. Designing Reusable Classes. *Journal of Object-Oriented Programming*, 1 (2), 22-35.

Jones, M. S., McKeown, G. P., Rayward-Smith, V. J., 2002. Distribution, Cooperation, and Hybridization for Combinatorial Optimization. In: *Optimization Software Class Libraries*. Kluwer.

Voß, S., Woodruff, D. L., 2002. Optimization Software Class Libraries. In: *Optimization Software Class Libraries*. Kluwer.

A CAD SYSTEM IN AUGMENTED REALITY APPLICATION

Pier Paolo Valentini[(a)], Davide Gattamelata[(b)], Eugenio Pezzuti[(c)]

[(a)]Università degli Studi di Roma Tor Vergata
[(b)] Università degli Studi di Roma Tor Vergata
[(c)] Università degli Studi di Roma Tor Vergata

[(a)]Valentini@ing.uniroma2.it, [(b)]Gattamelata@ing.uniroma2.it, [(c)]pezzuti@ing.uniroma2.it

ABSTRACT

In this paper an integration between a computer aided 3D modeller and an augmented reality environment is presented. The system is based on an high resolution web cam to acquire video stream from the real world and an electromagnetic tracking system (Flock of Bird by Ascension) which allows the user to interact with real and virtual objects in the augmented scene. The software to manage user interaction and data flow is implemented in Visual C++ and it makes use of the Artoolkit libraries, the OpenGL libraries and the Flock of Birds libraries. The purpose of the system is to speed up reverse engineering and prototyping processes, because the user can relate real object features in the scene to model its virtual entities or acquire geometrical features of existing parts. Moreover, the user can export the virtual models into CAD system or import external models to see how they fit in their real environment.

Keywords: Augmented Reality, CAD, motion tracking

1. INTRODUCTION

The Augmented Reality (AR) is an emerging field which deals with the combination of real world image and computer generated data. With an AR system, the user can extend the visual perception of the world, being supported by additional information and virtual objects. At present, most AR research is concerned with the use of live video imagery which is digitally processed and "augmented" by the addition of computer generated graphics. Many engineering applications are based on the video see-through system that is based on the use of one or two cameras which acquire an image stream from the real world. This stream is processed by a computer which produces an augmented image stream which is projected again to the user by means of a blind visor. The major aim of an augmented reality (AR) system is the need of a consistent registration between the real world and the virtual one, making the illusion that the virtual objects are part of the real scene (Bimber and Raskar, 2005). On the other hand, many scientists and technicians of the augmented reality pay attention on the integration process of virtual features in real scene without taking care of interactivity problems. Referring to some AR applications as the maintenance of aeronautical or mechanical equipments or surgical applications, the lack of interactivity seems not to be a problem. Indeed, for these purposes, the cheapness of an AR system to visualize mixed scenes, with low cost of implementation, appears the winning choice.

Moving the attention to the industrial field, according to many researchers, the AR systems seem to be the future of Computer Aided Design (CAD) technologies and Virtual Engineering. The idea is to develop an AR system which is able to support the design in engineering and science. The first step towards this target is that the designer has to be able to model the shapes of products in the augmented world, evaluating real time their interaction with the real world. This means that the user not only has to perceive the augmented environment in a visual way, but also has to interact with it communicating his ideas. For this purpose, the visual scene has to be provided with an additional tracking system that is able to record the position of a virtual pen (the pointer) that is used to model the shapes (Liverani, Amati, Carbone, Caligana, 2005). Scientific literature reports the usage of mainly two types of tracking devices: optical trackers (Fiorentino, 2003) and haptic system (Bruno, Caruso, Pina, Muzzupappa, 2007), (Vallino, 1998). The usage of an haptic device to track the position of a pointer in the scene produces extremely precise results, but the working space reached by the robotized arm is limited. In addition the movement of the user is not completely free. The usage of optical systems allows a wider working area with less precision. It is also affected by illumination problem and marker visibility requirements.

An adequate trade off among precision, working space and user friendliness is the use of magnetic tracking devices. For the investigation presented in this paper the authors have tested the *Flock of Birds* by Ascension (Ascension Technology Corporation, 2002).

The data from the tracking system have to be real time combined with the camera video flow. Due to the registration process the computer can align the CAD objects with the user point of view. At the same time the pc records the user's events like position of the pointer or CAD command activation. So the user is able to model virtual objects moving all around the scene.

2. SYSTEM CONFIGURATION AND SETUP

The implemented system (Figure 1) is made of input, processing and output devices. The input devices are a USB2.0 camera (Microsoft LifeCame VX-6000), the motion tracking system *Flock of Birds* (further details at: http://www.ascension-tech.com/products/flockofbirds.php) and a common pc keyboard. The first one supplies an artificial vision of the scene while the second provides the position of the pointer in the scene. The keyboard is used for key commands. The processing unit is a pc (Intel Pentium 4, CPU freq. 2,8GHz, motherboard Asus P5 series and graphic card NVIDIA 6600 series).

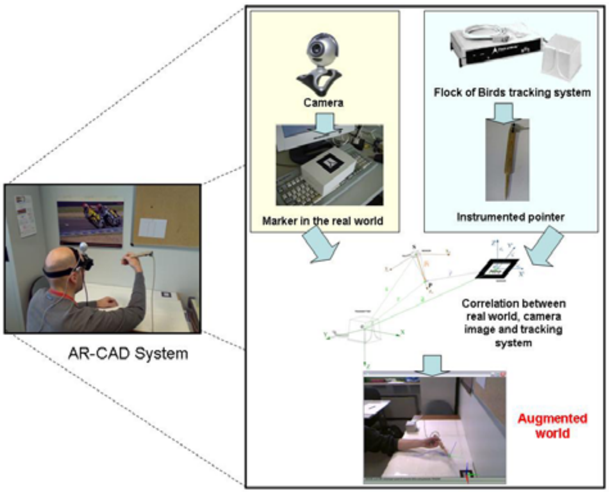

Camera
Marker in the real world
Flock of Birds tracking system
Instrumented pointer
Correlation between real world, camera image and tracking system
Augmented world
AR-CAD System

Figure 1: Overview of the implemented system.

The output device is a video see-trough display (eMagin Z800 3DVisor, Aspect ratio 4x3, Resolution SVGA 800 x 600 triad pixels per display (1.44 megapixels), contrast ratio >200:1, brightness >50cd/m2, 24 bit color, headtracking 360° horizontal, >60° vertical). The pc processes the input data and combines them to obtain the video data flow of the augmented scene.

The data flows coming from the input devices have to be synchronized and collimated by means of two relations: one between the real world and the camera and the other between the real world and the tracking system. The collimation between the virtual world and the real one involves some mathematical transformations. It is necessary to express the pointer tip position (*P* in Figure 2) in the reference frame attached to the marker and then apply the camera transformation to project the point according to the camera point of view.

With reference to Figure 2, the first transformation concerns the evaluation of position of *P* in the reference frame of the transmitter (O-*XYZ*) and it can be written as:

$$\{r\}=\{s\}+\left[M\left(\Psi,\theta,\varphi\right)\right]\cdot\{P_t\} \qquad (1)$$

where $\{r\}$ contains the coordinates of the tip *P* in the transmitter reference frame (*S-XYZ*); $\left[M\left(\Psi,\theta,\varphi\right)\right]$ is the rotation matrix built from the attitude angles $\left(\Psi,\theta,\varphi\right)$ of the sensor reference frame (S-$x_s y_s z_s$); $\{s\}$ is the translation vector from the origin of the transmitter reference frame to that of the sensor; $\{P_t\}$ is the vector which indicates the local position of the tip *P* in the sensor reference frame.

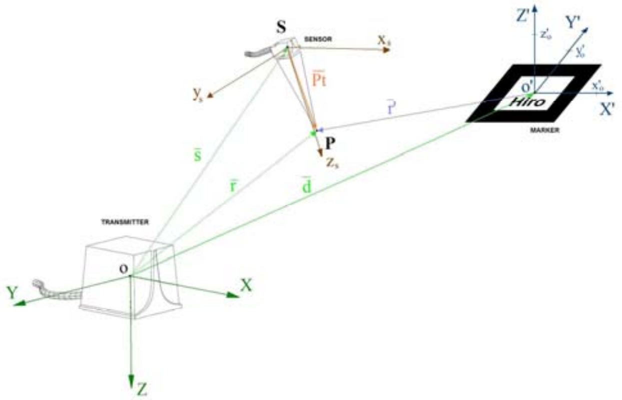


Figure 2: The collimation Process.

The second transformation concerns the computation of the coordinates $\{r'\}$ of pointer tip *P* in the reference frame of the marker (*o'-X'Y'Z'*). This transformation can be written as:

$$\{r'\}=-\{d\}+\left[T\left(\Psi',\theta',\varphi'\right)\right]\cdot\{r\} \qquad (2)$$

where $\{d\}$ is the vector between the origin of the marker reference frame an that of the transmitter; $\left[T\left(\Psi',\theta',\varphi'\right)\right]$ is the rotation matrix between the marker and the transmitter reference frames. The distance $\{d\}$ and the matrix $\left[T\left(\Psi',\theta',\varphi'\right)\right]$ are independent from the location of the pointer in the scene and have to be computed only at the beginning of the registration.

The registration process can be made by the user selecting with the pointer tip *P* the origin *O'* and two points on X', Y' axes of the reference frame of the marker.

In order to project the virtual object in the scene another transformation has to be applied. This transformation involves the computation of the relative position between marker and camera. It concerns the recognition of the marker features by pattern matching (Figure 3) and the computation of a distance vector and a rotation matrix.

These two entities can be used to render any virtual object (augmented contents) on the scene using the graphic pipeline typical of OperGL (Wright, Lipchak, 2004) (Figure 4).

The camera transformation is based on the calibration procedure, which aims to know the intrinsic and extrinsic parameters of the camera. They are necessary to link the pixel coordinates of an image point to the corresponding coordinates in the reference frame of the marker (Trucco, Verri, 1998) and it is supplied by the ARToolkit libraries (Lamb, 1999).

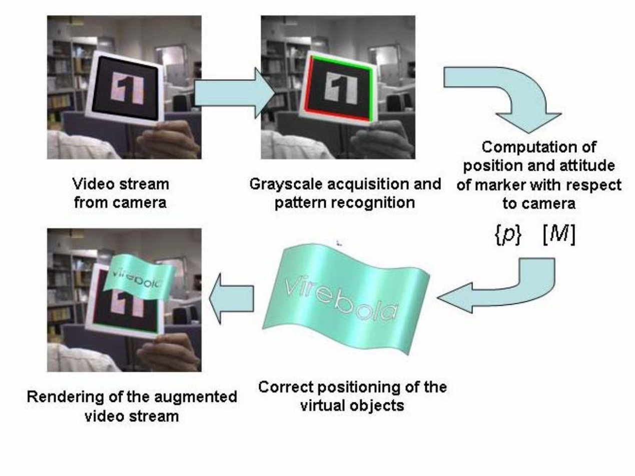


Figure 3. Pattern recognition pipeline.

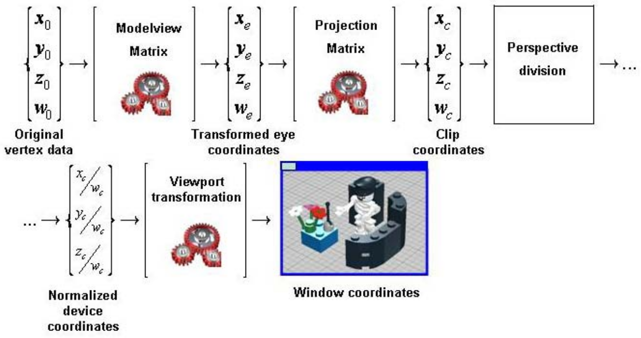


Figure 4: The render transformations pipeline.

3. DATA FLOW AND CAD ENGINE

The core of the proposed system is the *ad hoc* developed software AR-CAD 1.0. Its role is to manage the data flow coming from camera and sensor and to process the output video stream. Mainly there are three different data flows in the system (see Figure 5).

The first data flow comes from the camera and interests the image processing section of the software. Using the artoolkit method *ARGetTransMat()* the software computes the camera parameters transformation described in the previous section of the paper.

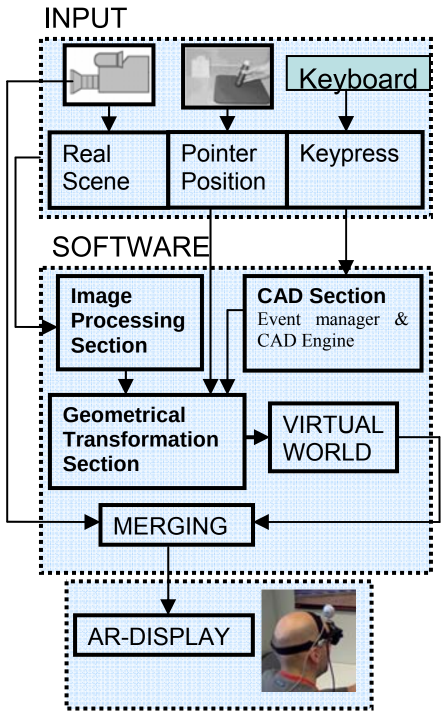
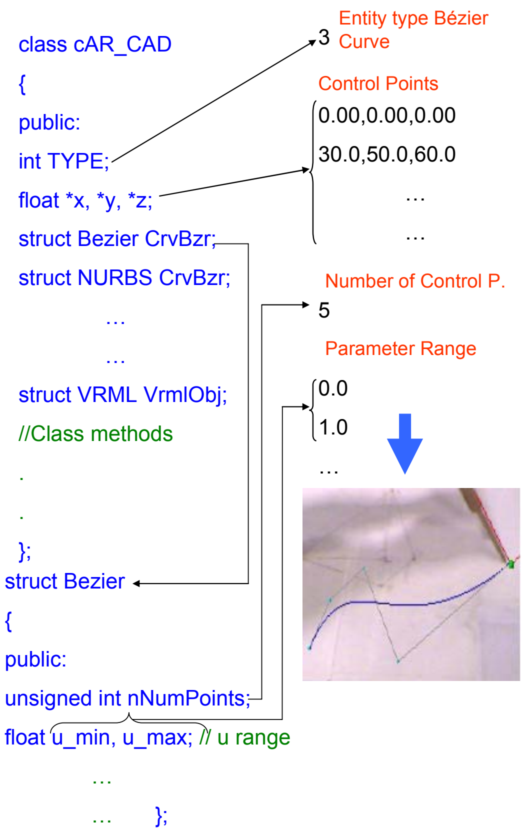


Figure 5 : Data Flow Scheme.

The second processes the data flow coming from the *Flock of Birds* and uses the geometrical transformation section already presented. In this way the equations (1) and (2) are used to compute the pointer tip coordinates in the reference frame of the marker.

The third data flow concerns the user's key-press events and is processed in the CAD section.

The CAD section is made of two parts: the modelling core and the events manager (see Figure 5). The modelling core is a set of classes and methods which are used to store and manipulate the virtual objects. The class *cAR_CAD* manages every geometrical entities (curves, lines, planes, surfaces and solid) in the 3D virtual space. When the user activates a CAD function, the event manager uses the corresponding methods of the library (*AR_Functions.h*) to generate an instance of the *cAR_CAD* class and to store the selected points in a right way.



```
class cAR_CAD
{
public:
int TYPE;
float *x, *y, *z;
struct Bezier CrvBzr;
struct NURBS CrvBzr;
        …
        …
struct VRML VrmlObj;
//Class methods
.
.
};
struct Bezier
{
public:
unsigned int nNumPoints;
float u_min, u_max; // u range
        …
        …    };
```

Figura 6: An example of how to use the cAR_CAD class to manage a Bézier curve.

Referring to the structure (Figure 6) each object is defined by an integer (named TYPE) which represents the type (line, polyline, parametric curve), then by three dynamically allocable arrays *x,y,z* to store the list of points and finally by the relative data structures. Depending on the entity to build, only one of the data structures is initialized. Parametric curves and parametric surfaces (like NURBS or Bézier curves and surfaces) have dedicated data structures. The solid entities use a common structure which is similar to the VRML one (Hartman and Wernecke, 1998).

Finally all the data flows converge in the virtual world section (see Figure 5) which elaborates the objects in the memory stack to project them in the augmented scene. The last operation is that of merging, in which the software render the virtual objects to the real world and send the augmented video frame to the AR-Display.

Two classes of objects have been implemented. The first one comprises the entities stored using only one instance of structure (line, polyline, parametric curves and surfaces by points selection). The second includes the entities stored with nested data structures and is used for building solid objects and surfaces by extrusion, sweep or loft.

With reference to Figure 7, the structure of the solid is defined starting from the basis sketch (green hexagon), which is stored in a curve data structure type.

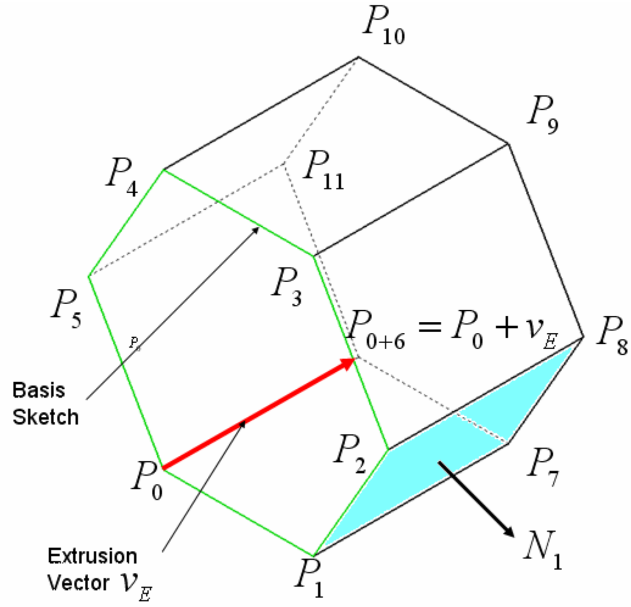


Figure 7: An example of nested data structures to build an extruded solid.

Following the *VRML* syntax in order to index the faces, the vertices of the extruded face are ordered in sequence. For an extrusion of a polygon of n vertices $(P_0,...,P_{n-1})$ we have a solid object of $2n$ vertices $(P_0,...,P_{n-1},P_n,...,P_{2n-1})$ with $n+2$ faces. The secondary vertices are calculated through the following expression:

$$P_{i+n} = P_i + \overrightarrow{v_E} \qquad (i = 0,..,n-1) \tag{3}$$

where $\overrightarrow{v_E}$ is the extrusion vector.

In order to draw the stored entities in the scene, the software uses the functions implemented in *AR_Functions.h*. These functions make use of the basic OpenGl methods *glBegin()* and *glVertex*`3f` using the mode *GL_LINE_STRIP*, to draw curve pieces or lines, and the *GL_POLYGON* to draw faces and tessellated surfaces [R. S. Wright et al. 2004].

Looking at the extrusion example of Figure 7, the graphic engine draws $n+2$ faces F_i $(i = 0..n+1)$. F_0 is the face of basis sketch and depends on the basis sketch vertices $(P_0,...,P_{n-1})$ and F_{n+1} is the extruded face and depends on the list of secondary vertices $(P_n,...,P_{2n-1})$. At last the n lateral faces F_k $(k = 1..n)$ which depend on the lists of four vertices $(P_k, P_{k-1}, P_{k-1+n}, P_{k+n})$. The vertices of lateral faces are ordered counterclockwise to have the face normal pointing outside the solid (see N_1 in Figure 8).

4. MODELING IN AUGMENTED REALITY

The geometric modeler in the AR-system has been implemented as a feature based builder, as it happens in many recent CAD programs. For this purpose, it is possible to classify the implemented functions in two categories: that of the sketch commands and that of the applied functions.

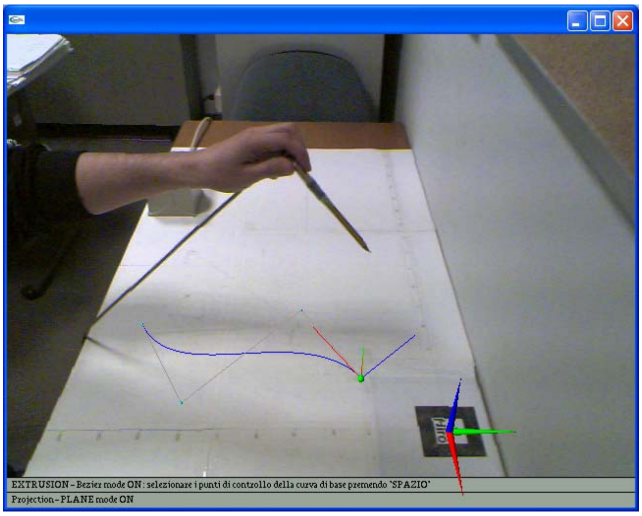

EXTRUSION - Bezier mode ON : selezionare i punti di controllo della curva di base premendo 'SPAZIO'
Projection - PLANE mode ON

Figure 8: Drawing of parametric Bézier Curve projected on the selected sketch plane.

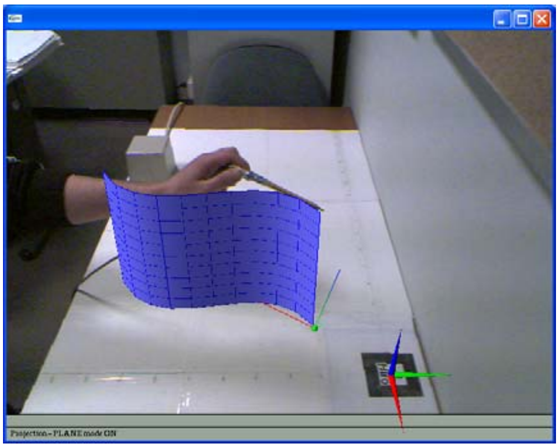

Projection - PLANE mode ON

Figure 9: Drawing of a surface by mean of curve extrusion

In the first category, we have elementary geometrical shapes and parametric plane curves. The **Figure** 8 reports the drawing action of a parametric curve projected on a sketch plane.

The category of applied functions to the sketch is composed from two subclasses, the class for surface construction and the class for solid modeling.

Figure 9 reports an example of surface construction by means of an extrusion of the Bézier curve in the previous example.

Solid objects can be built selecting a sketch and applying a specific function. **Figure** 10 reports two cases: an extrusion of a circle to model a cylinder and that of a closed profile.

The 3D entities can be also built selecting points in space without the use of sketch plane. An example of solid modeling of a sphere is reported in Figure 11 while an example of a generic surface is reported in Figure 12.

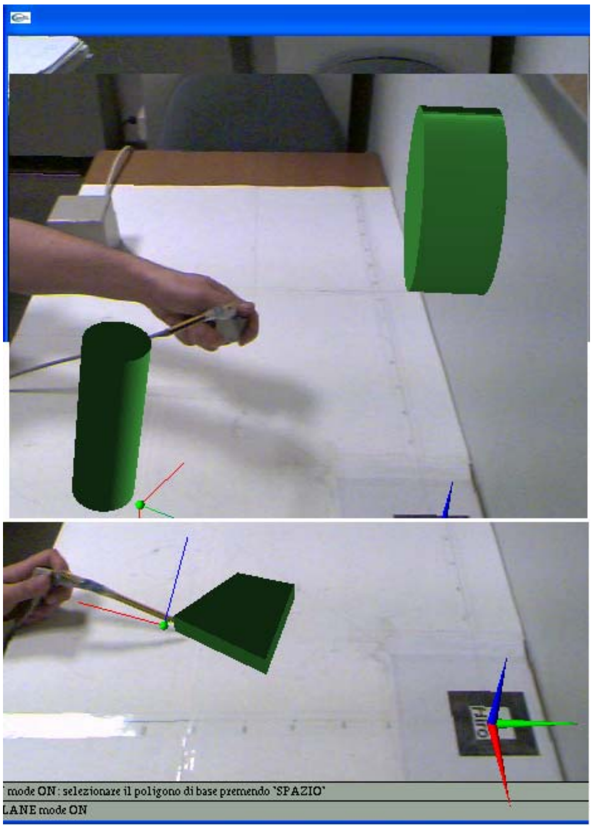

' mode ON : selezionare il poligono di base premendo 'SPAZIO'
LANE mode ON

Figure 10: Solid parts from interactive extrusion (two cylinders, on the top; a generic polygon extrusion, at the bottom).

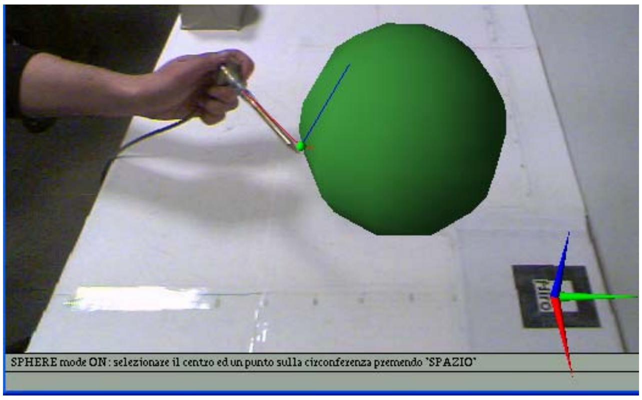

SPHERE mode ON : selezionare il centro ed un punto sulla circonferenza premendo 'SPAZIO'

Figure 11 Solid modeling of a sphere.

The major advantages of drawing entities referring directly to real points and real objects can be noted for the complex surface construction by means of points selection. (Figure 12).

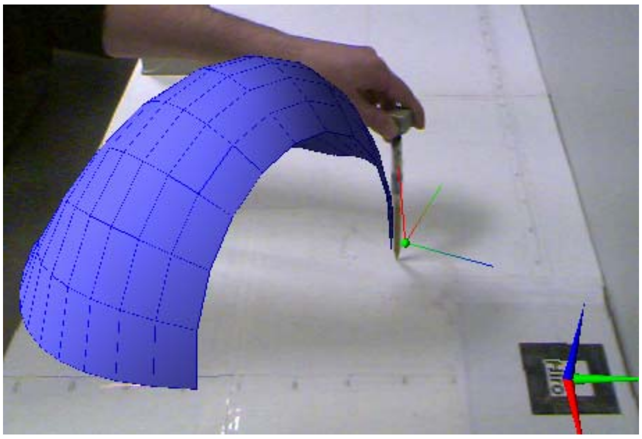


Figure 12 Surface Modeling.

5. CONCLUSIONS

In this paper an hardware and software setup has been presented in order to implement a modeling environment in augmented reality. The described architecture is able to speed up reverse engineering and prototyping tasks at a reasonable cost and with a good reliability and precision. The great advantage for the designer is that he can model and modify objects using an intuitive device (the magnetic pointer) and directly in the real world. The combined use of camera, pointer and head mounted displays have been managed by means of a specific software. This software play the role of an interface between the hardware devices (input and output) and the graphics engine. The modeling procedures are based on the OpenGL libraries. The use of a magnetic pointer has the advantage to widen the modeling space and make the picking operations simpler.

For engineering and industrial applications, one of the crucial aspects is the picking precision of the instrumented pointer. It is mainly affected by two factors. The first depends on the intrinsic instrument precision and software implementation. The pointer position is sensitive to the static resolution of the motion tracking system (about 0.5 mm for the position and about 0.1° for the orientation) and the acquisition sampling rate. The *FOB* is able to acquire up to 144 frames per second (fps), but it is forced to work at 30 fps, synchronized with the camera. A lower sampling rate is also justified considering the limited velocity of user's movements during modeling.

The second contribution to the system precision depends on user's limited accuracy during picking, which is subjective.

Moving the attention to the AR component of the system some improvements can be introduced for future work. The code concerning the modeling engine can be extended with new commands to speed up the modeling procedure. Moreover, the implementation of functions for advanced surface and solid editing can improve the modeling capabilities similarly to modern CAD environments.

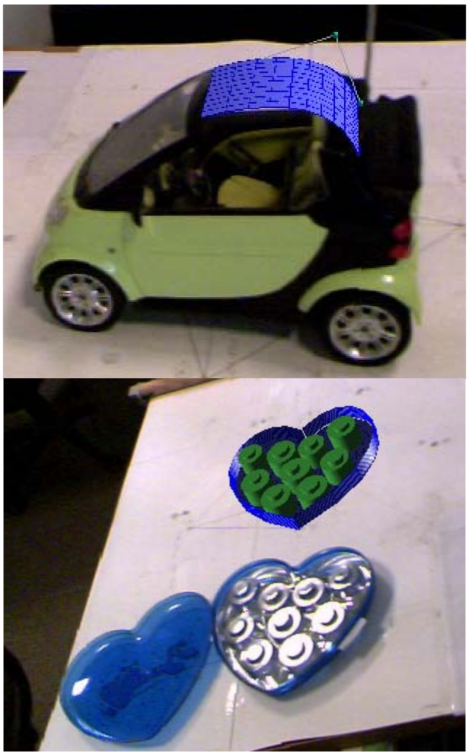


Figure 13: Reverse engineering of a toy car roof (on the top) and of a chocolate box.

In addition, the interactivity of the system can be improved introducing other devices like virtual glove or introducing a virtual keyboard. At last, to increase the illusion of the augmented scene, it is desirable to have a collimation between the illumination of the real scene and that of the virtual one.

REFERENCES

Ascension Technology Corporation, 2002. *Flock of Birds API reference guide*. Available from: ftp://ftp.ascensiontech.com/DRIVERS/WINDOWS_DRIVER/ [accessed 10 June 2008].

Bimber, O. and Raskar, R., 2005. *Spatial Augmented Reality Merging Real and Virtual Worlds 2005.* United States of America: A K Peters, Ltd.

Bruno, F., Caruso, F., Pina, M., Muzzupappa, M., 2007. Sviluppo di un tool di augmented reality di supporto alle tecniche di Reverse Engineering per contatto. *Proceedings of ADM Congress*. 6-8 Giugno 2007, Perugia (Italy).

Fiorentino, M., 2003. *Design estetico in mixed reality*. Phd Thesis, Politecnico di Bari, Italy.

Hartman, J. and Wernecke, J., 1998. *The VRML 2.0 Handbook*, United States of America: Silicon Graphics Inc.

Lamb, P., 1999, *Artoolkit.* Available from: http://www.hitl.washington.edu/artoolkit/ [accessed 10 June 2008]

Liverani, A., Amati, G., Carbone, L.,G. Caligana, G., 2005. Modellazione e ricostruzione video assistita di componenti. *Congresso congiunto ADM INGEGRAF*, Giugno 2005, Siviglia (Spain).

Trucco, E., Verri, A., 1998. *Introductory Techniques for 3-D Computer Vision*, New Jersey: Prentice Hall.

Vallino, J. R. 1998. *Interactive Augmented reality*. Phd Thesis, University of Rochester, New York.

Wright, R. S., Lipchak, B., 2004. *OpenGl Super Bible, third edition*. United States of America: Sams Publishing

SPHERICAL HARMONICS ALGORITHM FOR DYNAMIC LIGHT IN REAL TIME

Lien Muguercia Torres

Facultad 5 Entornos Virtuales
Universidad de las Ciencias Informáticas
Ciudad Habana, Cuba
lmuguercia@uci.cu

ABSTRACT

Traditionally, efficient shading and light mapping has been an obstacle in computer graphics. Reaching desired realism levels requires a high source consume and sometimes it doesn’t accomplish existing expectative.

The intention of this work is adding more realism to virtual scenes through dynamic light using GPU.

Starting off of the study of many algorithms, it proposes many algorithms to work whit light through shader (program used to determine the final surface properties of an object or image. This can include arbitrarily complex descriptions of light absorption and diffusion, texture mapping, reflection and refraction, shadowing, surface displacement and post-processing effects)but we’ll propound the Spherical Harmonics algorithms and implemented in OpenGL Shading Language (GLSL).

Keywords: GPU, shader, post-processing effects.

INTRODUCTION

The recent trend in graphics hardware has been to replace fixed functionality with programmability in areas that have grown exceedingly complex.

Two such areas are vertex processing and fragment processing. Vertex processing involves the operations that occur at each vertex, most notably transformation and lighting. Fragments are per-pixel data structures that are created by the rasterization of graphics primitives.

A fragment contains all the data necessary to update a single location in the frame buffer. Fragment processing consists of the operations that occur on a per-fragment basis, most notably reading from texture memory and applying the texture value(s) at each fragment.

With the OpenGL Shading Language, the fixed functionality stages for vertex processing and fragment processing have been augmented with programmable stages that can do everything the fixed functionality stages can doand a whole lot more. The OpenGL Shading Language allows application programmers to express the processing that occurs at those programmable points of the OpenGL pipeline.

The OpenGL Shading Language code that is intended for execution on one of the OpenGL programmable processors is called a shader. The term OpenGl shader is sometimes used to differentiate a shader written in the OpenGL Shading Language from a shader written in another shading language such as RenderMan.

In this article I explore how the OpenGL Shading Language can help us implement such models so that they can execute at interactive rates on programmable graphics hardware. We look at some lighting models that provide more flexibility and give more realistic results than those built into OpenGL's fixed functionality rendering pipeline. Much has been written on the topic of lighting in computer graphics. We only examine a few methods an propound the implementation of one . Hopefully, you'll be inspired to try implementing some others on your own.

BASIC

Why shader?

By exposing support for traditional rendering mechanisms, OpenGL has evolved to serve the needs of a fairly broad set of applications. If your particular application was well served by the traditional rendering model presented by OpenGL, you may never need to write shaders. But if you have ever been frustrated because OpenGL did not allow you to define area lights, or because lighting calculations are performed per-vertex rather than per-fragment or, if you have run into any of the many limitations of the traditional OpenGL rendering model, you may need to write your own OpenGL shader.

With each new generation of graphics hardware, more complex rendering techniques can be implemented as

OpenGL shaders and can be used in real-time rendering applications. Here's a brief list of what's possible with OpenGL shaders:

- Increasingly realistic lighting effects area lights, soft shadows
- Advanced rendering effects global illumination, ray-tracing,
- Animation effects key frame interpolation, particle systems, procedurally defined motion
- User programmable anti-aliasing methods
- General computation sorting, mathematical modelling, fluid dynamics, and so on

Many of these techniques have been available before now only through software implementations. If they were at all possible through OpenGL, they were possible only in a limited way. The fact that these techniques can now be implemented with hardware acceleration provided by dedicated graphics hardware means that rendering performance can be increased dramatically and at the same time the CPU can be off-loaded so that it can perform other tasks.

LIGTHING

In the real world, we see things because they reflect light from a light source or because they are light sources themselves. In computer graphics, just as in real life, we won't be able to see an object unless it is illuminated or emits light. To generate more realistic images, we need to have more realistic models for illumination, shadows, and reflection than those we've discussed so far.

Hemisphere Lighting

We know at the fixed functionality lighting model built into OpenGL and developed shader code to mimic the fixed functionality behavior. However, this model has a number of flaws, and these flaws become more apparent as we strive for more realistic rendering effects. One problem is that objects in a scene do not typically receive all their illumination from a small number of specific light sources. Interreflections between objects often have noticeable and important contributions to objects in the scene. The traditional computer graphics illumination model attempts to account for this phenomena through an ambient light term. However, this ambient light term is usually applied equally across an object or an entire scene. The result is a flat and unrealistic look for areas of the scene that are not affected by direct illumination.

Another problem with the traditional illumination model is that light sources in real scenes are not point lights or even spotlightsthey are area lights. Consider the indirect light coming in from the window and illuminating the floor and the long fluorescent light bulbs behind a rectangular translucent panel. For an even more common case, consider the illumination outdoors on a cloudy day. In this case, the entire visible hemisphere is acting like an area light source. In several presentations and tutorials, Chas Boyd, Dan Baker, and Philip Taylor of Microsoft described this situation as Hemisphere Lighting and discussed how to implement it in DirectX. Let's look at how we might create an OpenGL shader to simulate this type of lighting environment.

The idea behind hemisphere lighting is that we model the illumination as two hemispheres. The upper hemisphere represents the sky, and the lower hemisphere represents the ground. A location on an object with a surface normal that points straight up gets all of its illumination from the upper hemisphere, and a location with a surface normal pointing straight down gets all of its illumination from the lower hemisphere (see Figure 1). By picking appropriate colors for the two hemispheres, we can make the sphere look as though locations with normals pointing up are illuminated and those with surface normals pointing down are in shadow.

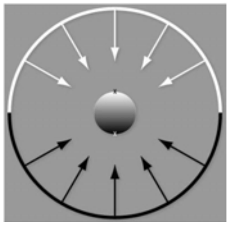


Figure 1: A sphere illuminated using the hemisphere lighting model

To compute the illumination at any point on the surface, we must compute the integral of the illumination received at that point:

Color = a · SkyColor + (1 - a) · GroundColor
where
a = 1.0 - (0.5 · sin(q)) for q < 90°
a = 0.5 · sin(q) for q > 90°
q = angle between surface normal and north pole direction

But we can actually calculate a in another way that is simpler but roughly equivalent:

a = 0.5 + (0.5 · cos(q))

This approach eliminates the need for a conditional. Furthermore, we can easily compute the cosine of the angle between two unit vectors by taking the dot product of the two vectors. This is an example of what Jim Blinn likes to call "the ancient Chinese art of chi ting." In computer graphics, if it looks good enough, it is good enough. It doesn't really matter whether your calculations are physically correct or a colossal cheat.

One of the issues with this model is that it doesn't account for self-occlusion. Regions that should really be in shadow because of the geometry of the model appear too bright.

Image-Based Lighting

If we're trying to achieve realistic lighting in a computer graphics scene, why not just use an environment map for the lighting? This approach to illumination is called Image-Based Lighting; it has been popularized in recent years by researcher Paul Debevec at the University of Southern California. Churches and auditoriums may have dozens of light sources on the ceiling. Rooms with many windows also have complex lighting environments. It is often easier and much more efficient to sample the lighting in such environments and store the results in one or more environment maps than it is to simulate numerous individual light sources.

The steps involved in image-based lighting are:

- Use a Light Probe (e.g., a reflective sphere) to capture (e.g., photograph) the illumination that occurs in a real-world scene. The captured omnidirectional, high-dynamic range image is called a Light Probe Image.
- Use the light probe image to create a representation of the environment (e.g., an environment map).
- Place the synthetic objects to be rendered inside the environment.
- Render the synthetic objects by using the representation of the environment created in step 2.

On his Web site (http://www.debevec.org/), Debevec offers a number of useful things to developers. For one, he has made available a number of images that can be used as high-quality environment maps to provide realistic lighting in a scene. These images are high dynamic range (HDR) images that represent each color component with a 32-bit floating-point value. Such images can represent a much greater range of intensity values than can 8-bit-per-component images. For another, he makes available a tool called HDRShop that manipulates and transforms these environment maps. Through links to his various publications and tutorials, he also provides step-by-step instructions on creating your own environment maps and using them to add realistic lighting effects to computer graphics scenes.

Following Debevec's guidance, I purchased a 2-inch chrome steel ball from McMaster-Carr Supply Company (http://www.mcmaster.com). We used this ball to capture a light probe image from the center of the square outside our office building in downtown Fort Collins, Colorado. We then used HDRShop to create a lat-long environment map and a cube map of the same scene. The cube map and lat-long map can be used to perform environment mapping. That shader simulated a surface with an underlying base color and diffuse reflection characteristics that was covered by a transparent mirror-like layer that reflected the environment flawlessly.

We can simulate other types of objects if we modify the environment maps before they are used. A point on the surface that reflects light in a diffuse fashion reflects light from all the light sources that are in the hemisphere in the direction of the surface normal at that point. We can't really afford to access the environment map a large number of times in our shader. What we can do instead is similar to what we discussed for hemisphere lighting. Starting from our light probe image, we can construct an environment map for diffuse lighting. Each texel in this environment map will contain the weighted average (i.e., the convolution) of other texels in the visible hemisphere as defined by the surface normal that would be used to access that texel in the environment.

Again, HDRShop has exactly what we need. We can use HDRShop to create a lat-long image from our original light probe image. We can then use a command built into HDRShop that performs the necessary convolution. This operation can be time consuming, because at each texel in the image, the contributions from half of the other texels in the image must be considered. Luckily, we don't need a very large image for this purpose. The effect is essentially the same as creating a very blurry image of the original light probe image. Since there is no high frequency content in the computed image, a cube map with faces that are 64 x 64 or 128 x 128 works just fine.

A single texture access into this diffuse environment map provides us with the value needed for our diffuse reflection calculation. What about the specular contribution? A surface that is very shiny will reflect the illumination from a light source just like a mirror. A single point on the surface reflects a single point in the environment. For surfaces that are rougher, the highlight defocuses and spreads out. In this case, a single point on the surface reflects several points in the environment, though not the whole visible hemisphere like a diffuse surface. HDRShop lets us blur an environment map by providing a Phong exponenta degree of shininess. A value of 1.0 convolves the environment map to simulate diffuse reflection, and a value of 50 or more convolves the environment map to simulate a somewhat shiny surface.

The shaders that implement these concepts end up being quite simple and quite fast. In the vertex shader, all that is needed is to compute the reflection direction at each vertex. This value and the surface normal are sent to the fragment shader as varying variables. They are interpolated across each polygon, and the interpolated values are used in the fragment shader to access the two environment maps in order to obtain the diffuse and the specular components. The values obtained from the environment maps are combined with the object's base color to arrive at the final color for the fragment.

The ÜberLight Shader

We've discussed lighting algorithms that simulate the effect of global illumination for more realistic lighting effects. Traditional point, directional, and spotlights can be used in conjunction with these global illumination effects. However, the traditional light sources leave a lot to be desired in terms of their flexibility and ease of use.

Ronen Barzel of Pixar Animation Studios wrote a paper in 1997 that described a much more versatile lighting model specifically tailored for the creation of computer-generated films. This lighting model has so many features and controls compared to the traditional graphics hardware light source types that its RenderMan implementation became known as the "überlight" shader (i.e., the lighting shader that has everything in it except the proverbial kitchen sink). Larry Gritz wrote a public domain version of this shader that was published in Advanced RenderMan: Creating CGI for Motion Pictures, which he coauthored with Tony Apodaca. A Cg version of this shader was published by Fabio Pellacini and Kiril Vidimice of Pixar in the book GPU Gems, edited by Randima Fernando. The full-blown überlight shader has been used successfully in a variety of computer-generated films, including Toy Story, Monsters, Inc., and Finding Nemo. Because of the proven usefulness of the überlight shader, this section looks at how to implement its essential features in the OpenGL Shading Language.

In movies, lighting helps to tell the story in several different ways. Sharon Calahan gives a good overview of this process in the book Advanced RenderMan: Creating CGI for Motion Pictures. This description includes five important fundamentals of good lighting design that were derived from the book Matters of Light & Depth by Ross Lowell:

- Directing the viewer's eye
- Creating depth
- Conveying time of day and season
- Enhancing mood, atmosphere, and drama
- Revealing character personality and situation

Because of the importance of lighting to the final product, movies have dedicated lighting designers. To light computer graphics scenes, lighting designers must have an intuitive and versatile lighting model to use.

For the best results in lighting a scene, it is crucial to make proper decisions about the shape and placement of the lights. For the überlight lighting model, lights are assigned a position in world coordinates. The überlight shader uses a pair of superellipses to determine the shape of the light. A superellipse is a function that varies its shape from an ellipse to a rectangle, based on the value of a roundness parameter. The superellipse function is defined as

$$\left(\frac{x}{a}\right)^{\frac{2}{d}} + \left(\frac{y}{b}\right)^{\frac{2}{d}} = 1$$

As the value for d nears 0, this function becomes the equation for a rectangle, and when d is equal to 1, the function becomes the equation for an ellipse. Values in between create shapes in between a rectangle and an ellipse, and these shapes are also useful for lighting. This is referred to in the shader as barn shaping since devices used in the theater for shaping light beams are referred to as barn doors.

It is also desirable to have a soft edge to the light, in other words, a gradual drop-off from full intensity to zero intensity. We accomplish this by defining a pair of nested superellipses. Inside the innermost superellipse, the light has full intensity. Outside the outermost superellipse, the light has zero intensity. In between, we can apply a gradual transition by using the smoothstep function.

Two more controls that add to the versatility of this lighting model are the near and far distance parameters, also known as the cuton and cutoff values. These define the region of the beam that actually provides illumination (see Figure 2). Again, smooth transition zones are desired so that the lighting designer can control the transition. Of course, this particular control has no real-world analogy, but it has proved to be useful for softening the lighting in a scene and preventing the light from reaching areas where no light is desired.

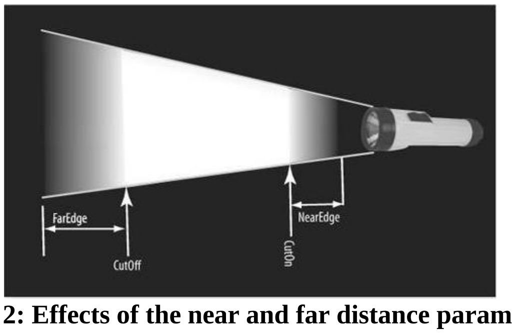


Figure 2: Effects of the near and far distance parameters for the überlight shader

The überlight shader as described by Barzel and Gritz actually has several additional features. It can support multiple lights, but our example shader showed just one for simplicity. The key parameters can be defined as arrays, and a loop can be executed to perform the necessary computations for each light source. In the following chapter, we show how to add shadows to this shader.

Lighting with Spherical Harmonics

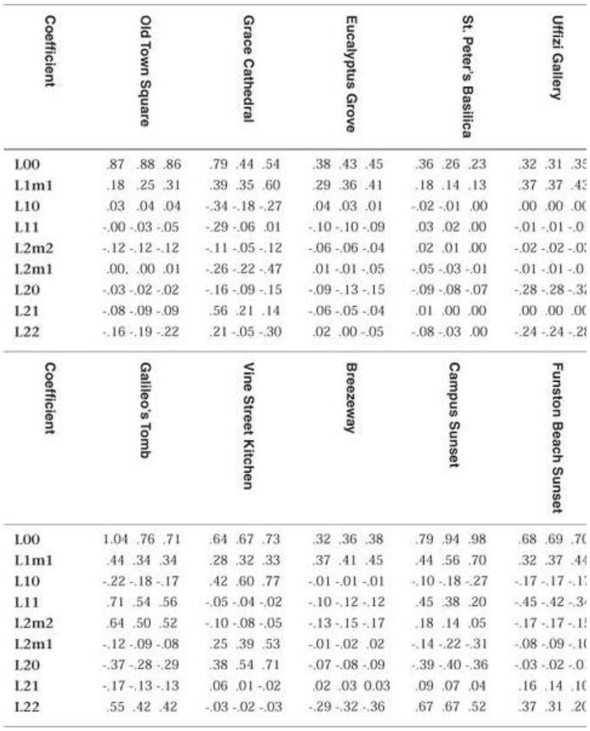
In 2001, Ravi Ramamoorthi and Pat Hanrahan presented a method that uses spherical harmonics for computing the diffuse lighting term. This method reproduces accurate diffuse reflection, based on the content of a light probe image, without accessing the light probe image at runtime.

The light probe image is pre-processed to produce coefficients that are used in a mathematical representation of the image at runtime. The mathematics behind this approach is beyond the scope of this book (see the references at the end of this chapter if you want all the details). Instead, we lay the necessary groundwork for this shader by describing the underlying mathematics in an intuitive fashion. The result is remarkably simple, accurate, and realistic, and it can easily be codified in an OpenGL shader. This technique has already been used successfully to provide real-time illumination for games and has applications in computer vision and other areas as well.

Spherical harmonics provides a frequency space representation of an image over a sphere. It is analogous to the Fourier transform on the line or circle. This representation of the image is continuous and rotationally invariant. Using this representation for a light probe image, Ramamoorthi and Hanrahan showed that you could accurately reproduce the diffuse reflection from a surface with just nine spherical harmonic basis functions. These nine spherical harmonics are obtained with constant, linear, and quadratic polynomials of the normalized surface normal.

Intuitively, we can see that it is plausible to accurately simulate the diffuse reflection with a small number of basis functions in frequency space since diffuse reflection varies slowly across a surface. With just nine terms used, the average error over all surface orientations is less than 3% for any physical input lighting distribution. With Debevec's light probe images, the average error was shown to be less than 1% and the maximum error for any pixel was less than 5%.

Each spherical harmonic basis function has a coefficient that depends on the light probe image being used. The coefficients are different for each colour channel, so you can think of each coefficient as an RGB value. A preprocessing step is required to compute the nine RGB coefficients for the light probe image to be used. Ramamoorthi makes the code for this pre-processing step available for free on his Web site. I used this program to compute the coefficients for all the light probe images in Debevec's light probe gallery as well as the Old Town Square light probe image and summarized the results in Figure 3.



Coefficient	Old Town Square	Grace Cathedral	Eucalyptus Grove	St. Peter's Basilica	Uffizi Gallery
L00	.87 .88 .86	.79 .44 .54	.38 .43 .45	.36 .26 .23	.32 .31 .3
L1m1	.18 .25 .31	.39 .35 .60	.29 .36 .41	.18 .14 .13	.37 .37 .4
L10	.03 .04 .04	-.34 -.18 -.27	.04 .03 .01	-.02 -.01 .00	.00 .00 .0
L11	-.00 -.03 -.05	-.29 -.06 .01	-.10 -.10 -.09	.03 .02 .00	-.01 -.01 -.0
L2m2	-.12 -.12 -.12	-.11 -.05 -.12	-.06 -.06 -.04	.02 .01 .00	-.02 -.02 -.0
L2m1	.00. .00 .01	-.26 -.22 -.47	.01 -.01 -.05	-.05 -.03 -.01	-.01 -.01 -.0
L20	-.03 -.02 -.02	-.16 -.09 -.15	-.09 -.13 -.15	-.09 -.08 -.07	-.28 -.28 -.3
L21	-.08 -.09 -.09	.56 .21 .14	-.06 -.05 -.04	.01 .00 .00	.00 .00 .0
L22	-.16 -.19 -.22	.21 -.05 -.30	.02 .00 -.05	-.08 -.03 .00	-.24 -.24 -.2

Coefficient	Galileo's Tomb	Vine Street Kitchen	Breezeway	Campus Sunset	Funston Beach Sunset
L00	1.04 .76 .71	.64 .67 .73	.32 .36 .38	.79 .94 .98	.68 .69 .7
L1m1	.44 .34 .34	.28 .32 .33	.37 .41 .45	.44 .56 .70	.32 .37 .4
L10	-.22 -.18 -.17	.42 .60 .77	-.01 -.01 -.01	-.10 -.18 -.27	-.17 -.17 -.1
L11	.71 .54 .56	-.05 -.04 -.02	-.10 -.12 -.12	.45 .38 .20	-.45 -.42 -.3
L2m2	.64 .50 .52	-.10 -.08 -.05	-.13 -.15 -.17	.18 .14 .05	-.17 -.17 -.1
L2m1	-.12 -.09 -.08	.25 .39 .53	-.01 -.02 .02	-.14 -.22 -.31	-.08 -.09 -.1
L20	-.37 -.28 -.29	.38 .54 .71	-.07 -.08 -.09	-.39 -.40 -.36	-.03 -.02 -.0
L21	-.17 -.13 -.13	.06 .01 -.02	.02 .03 0.03	.09 .07 .04	.16 .14 .1
L22	.55 .42 .42	-.03 -.02 -.03	-.29 -.32 -.36	.67 .67 .52	.37 .31 .2

Figure 3: Spherical harmonic coefficients for light probe images

The equation for diffuse reflection using spherical harmonics is

$$\text{Diffuse} = c_1 L_{22} (x^2 - y^2) + c_3 L_{20} z^2 + c_4 L_{20} - c_5 L_{20} + 2c_1 (L_{2-2}\, xy + L_{21}\, xz + L_{2-1}\, yz) + 2c_2 (L_{11}\, x + L_{1-1}\, y + L_{10}\, z$$

(1)

The constants c1c5 result from the derivation of this formula and are shown in the vertex shader code in Listing 1. The L coefficients are the nine basis function coefficients computed for a specific light probe image in the preprocessing phase. The x, y, and z values are the coordinates of the normalized surface normal at the point that is to be shaded. Unlike low dynamic range images (e.g., 8 bits per color component) that have an implicit minimum value of 0 and an implicit maximum value of 255, high dynamic range images represented with a floating-point value for each color component don't contain well-defined minimum and maximum values.

The minimum and maximum values for two HDR images may be quite different from each other, unless the same calibration or creation process was used to create both images. It is even possible to have an HDR image that contains negative values. For this reason, the vertex shader contains an overall scaling factor to make the final effect look right.

The vertex shader that encodes the formula for the nine spherical harmonic basis functions is actually quite simple. When the compiler gets hold of it, it becomes simpler still. An optimizing compiler typically reduces all the operations involving constants. The resulting code is quite efficient because it contains a relatively small number of addition and multiplication operations that involve the components of the surface normal.

```
varying vec3 DiffuseColor;
uniform float ScaleFactor;

const float C1 = 0.429043;
const float C2 = 0.511664;
const float C3 = 0.743125;
const float C4 = 0.886227;
const float C5 = 0.247708;

// Constants for Old Town Square 
lighting
const vec3 L00  = vec3( 0.871297, 
0.875222,  0.864470);
const vec3 L1m1 = vec3( 0.175058, 
0.245335,  0.312891);
const vec3 L10  = vec3( 0.034675, 
0.036107,  0.037362);
const vec3 L11  = vec3(-0.004629, -
0.029448, -0.048028);
const vec3 L2m2 = vec3(-0.120535, -
0.121160, -0.117507);
const vec3 L2m1 = vec3( 0.003242, 
0.003624,  0.007511);
const vec3 L20  = vec3(-0.028667, -
0.024926, -0.020998);
const vec3 L21  = vec3(-0.077539, -
0.086325, -0.091591);
const vec3 L22  = vec3(-0.161784, -
0.191783, -0.219152);
```

```
void main()
{
    vec3 tnorm    = normal-
ize(gl_NormalMatrix * gl_Normal);

    DiffuseColor =  C1 * L22 * 
(tnorm.x * tnorm.x - tnorm.y * 
tnorm.y) +
                    C3 * L20 * 
tnorm.z * tnorm.z +
                    C4 * L00 -
                    C5 * L20 +
                    2.0 * C1 * L2m2 * 
tnorm.x * tnorm.y +
                    2.0 * C1 * L21  * 
tnorm.x * tnorm.z +
                    2.0 * C1 * L2m1 * 
tnorm.y * tnorm.z +
                    2.0 * C2 * L11  * 
tnorm.x +
                    2.0 * C2 * L1m1 * 
tnorm.y +
                    2.0 * C2 * L10  * 
tnorm.z;

    DiffuseColor *= ScaleFactor;

    gl_Position = ftransform();
}
```

Listing 1 Vertex shader for spherical harmonics lighting

```
varying vec3 DiffuseColor;

void main()
{
    gl_FragColor = vec4(DiffuseColor, 
1.0);
}
```

Listing 2 Fragment shader for spherical harmonics lighting

Once again, our fragment shader has very little work to do. Because the diffuse reflection typically changes slowly, for scenes without large polygons we can reasonably compute it in the vertex shader and interpolate it during rasterization. As with hemispherical lighting, we can add procedurally defined point, directional, or spotlights on top of the spherical harmonics lighting to provide more illumination to important parts of the scene.

Coefficients for some of Paul Debevec's light probe images provide even greater color variations. We could make the diffuse lighting from the spherical harmonics computation more subtle by blending it with the object's base color.

CONCLUSION

Now the programmable graphics hardware has freed us from the shackles of the traditional hardware lighting equations, we are free to implement and experiment with a variety of new techniques. Some of the techniques we explored are both faster and more realistic than the traditional methods.

Such light probe images can either be preprocessed and used to compute spherical harmonic basis function coefficients that can be used for simple and high-performance lighting.

REFERENCES

Ramamoorthi, Ravi, and P. Hanrahan, 2001, *An Efficient Representation for Irradiance Environment Maps*, Computer Graphics (SIGGRAPH 2001 Proceedings), pp. 497500.

Baldwin, Dave, October 2001, *OpenGL 2.0 Shading Language White Paper*, Version 1.0, 3Dlabs.

ATI developer Web site. http://www.ati.com/developer

Akenine-Möller, Tomas, and E. Haines, 2002, *Real-Time Rendering, Second Edition*, AK Peters, Ltd., Natick, Massachusetts. http://www.realtimerendering.com

BIOGRAPHIES

Wesley, A., 2006, *OpenGL Shading Language 2nd Edition*. Orange Book: Addison Wesley Professional

A NEW HAPTIC-BASED TOOL FOR TRAINING IN MEDICINE

Tommaso Ingrassia[(a)], **Vincenzo Nigrelli**[(b)]

[(a) (b)]Università degli Studi di Palermo – Dipartimento di Meccanica - ITALY

[(a)] ingrassia@dima.unipa.it, [(b)]nigrelli@dima.unipa.it

ABSTRACT

This paper describes a new methodology that, making use of a haptic device, can simulate the palpation, a diagnostic manoeuvre aiming to verify the condition of internal organs or anatomical formations.

In the developed application, that has the purpose to pick out an anatomical formation and understand its characteristics, the palpation of a soft tissue has been taken in consideration. Particularly the fingertip, the skin and an anatomical formation have been simulated.

The user, handling the haptic system at disposal can feel the contact with the skin but also perceiving the presence, the shape and the dimension of the subcutaneous formation (invisible to the operator), that has been modelled as a rigid sphere (like a nodule).

The evaluation of the skin deformations, following from the palpation, has been performed through a mass-spring based algorithm, which allows to obtain results in real time.

Keywords: haptic, palpation simulation, virtual reality

1. INTRODUCTION

The sense of the touch is a mechanical stressing of the skin, produced by the physical contact with an object. Such stress depends on the superficial physical properties of the object and how it is touched on. Haptic technology tries to reproduce the sense of the touch in the contact with virtual objects. This advanced methodology allows to increase the reality feeling of a virtual simulation.

Haptic methodology has initially been developed as extension of the virtual reality (VR) simulations. Subsequently, because of not very large spreading of the VR systems, it has been employed in autonomous way with other typologies of applications.

The perception of the contact is reproduced through a suitable peripheral able to produce a force-feedback depending on the properties of the virtual objects.

Obviously it is simpler reproducing the mechanical properties of rigid bodies than simulating the mechanical performances of deformable bodies. In these last cases it needs referring to more sophisticated models of numerical analysis. The most used techniques are based on the finite element method (FEM), the boundary element method (BEM) and the mass spring model (MSM). Each method has some particular advantages and disadvantages, but in general it is possible to state that techniques based on FEM and BEM, even if can give more accurate results, need long times for data processing, whereas with the MSM technique real-time results can obtained.

In the research activity here presented it has been dealt with the problem of the development of a medical training application. An innovative methodology for the palpation, a diagnostic medical technique, has been set up.

Thanks to the developed application the user, by using the available force-feedback device, can moving a pointer, representing the fingertip, and can feel the contact with a surface representing a part of the human skin. Moreover he can also perceive the presence and the shape of a subcutaneous nodule.

Although the first prototype of this application has been implemented in a simplified way, by modelling the human skin like a planar quadrangular surface and a subcutaneous formation as a rigid sphere, the proposed methodology seems to be very powerful and presents many interesting starting points for the development of similar applications.

The mass spring model has been used to evaluate the forces; the objects surfaces have been represented through an implicit geometric formulation. In the setting up of the application, entirely developed in C++ language, OpenHaptics and OpenGL libraries have been used for the haptic and graphic rendering respectively.

2. STATE OF ARTS

Haptic methodology has been applied largely to several fields like the medical training and education (Wang et al. 2007), the reverse engineering (Yang et al. 2005), the mechanical modelling (Liu et al. 2004) (Ma et al. 2004), the artistic sculpture (Gao et al. 2006), the aerospace planning (Harris et al. 2004), etc.

In the surgical sector this technology allows to simulate in very complete way many operations on the human body that, generally, require significant tactile skills.

The more advanced methods of numerical analysis, like the boundary elements method (Wang et al. 2007) and the finite element method, allow to evaluate the forces in very accurate way so, consequently, to perform very realistic simulation, even if the complexity of calculation does not allow real time data processing.

The possibility to perform also real time simulations has led to the development of many training applications allowing doctors and surgeons to gain useful experience in very difficult medical techniques like laparoscopic and endoscopic (fig. 1) operations (Montgomery et al. 2002) (Okamura 2004).

Other very interesting examples of haptic applications in the medical field are the systems simulating the epidural injections (Dang et al. 2001), the educational applications to perceive in realistic way the contact with the anatomical parts of the human body (Heinrichs et al. 2000), the training systems for the positioning of catheters (Basdogan et al. 2001).

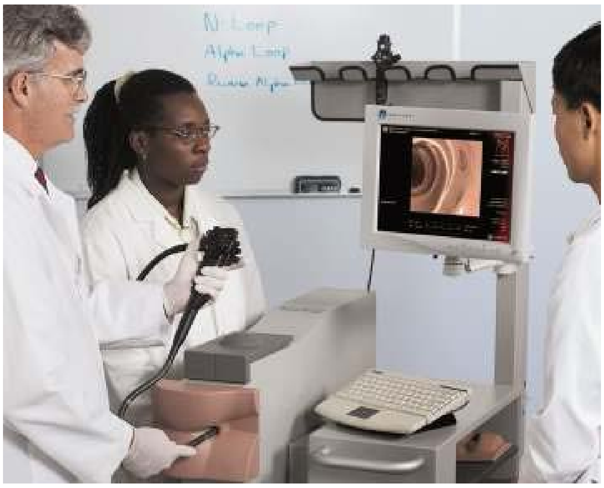


Figure 1: Haptic based device for endoscopic simulations

There are also advanced applications that make use of particular instrumentations like the force-feedback gloves (Bouzit et al. 2002), and the virtual reality environments to simulate surgical operations (Tendick et al. 2000) or to train doctors for the diagnosis of particular pathologies like the prostate cancer (Burdea et al. 1999).

With regard to the use of the haptic technology as an aid in the training for palpatory diagnosis, some research activities (Delingette 1998) (Langrana 1997) have already been developed, but nowadays not any proposed solution has been yet largely used in a very efficient way, for this reason it is opportune to continue in the research of very powerful and multi-purpose solutions, given that the palpation training represents a very important stage in the doctors' education and could be made more efficient through the use of this methodology.

3. PALPATORY TECHNIQUE

Palpation is a diagnostic manoeuvre performed with one or two hands, but also with the fingers (fig. 2), that is adopted to verify the state of internal organs or anatomical formations and their mutual positioning.

The most important information that can be got with palpatory diagnosis are related to the shape, the volume, the dimensions and the consistence of the internal organs, the characteristics of their surface (smooth, wrinkled, etc.), the temperature, the tenderness, the spontaneous and provoked mobility.

It is so evident the most important factor in the diagnosis through this simple technique, is a good training phase; a skilled doctor can, for instance, understand the difference between a stiffening and a volume increasing, or which dimension of a nodule represents a critical state.

Figure 2 - Example of palpating

The kind and quality of the information gained through this technique, therefore, depend on the skilfulness of the doctor, that can perceive the presence of anomalies by touching the interesting zone and feeling the changes of the mechanical behaviour as a consequence of the induced stressing.

It is clear that, in particular conditions, the touching sense represents a very important diagnostic method able to give basic information about many physical problems.

The mechanical response obtained as a consequence of the palpation is quite complex, in fact it is the reaction to a distributed load on a small area with not linear mechanical characteristics. Moreover this behaviour changes depending on the zone in examination: palpating a soft tissue differs from the analysis of an area in which a bony structure is present; this last, in fact, has very different mechanical characteristics in comparison with other organic tissues.

To simplify the application here introduced it has been studied the palpation on soft tissues with the purpose to identify an anatomical formation, perceiving its shape and stiffness.

4. TOOLS AND METHODS FOR THE HAPTIC APPLICATION DEVELOPMENT

In a practical way palpation is performed touching the skin surface with the fingertips. For this reason, in the developed application, a finger-based haptic device should be used.

In this job, nevertheless, a hand-based system, the Phantom Omni (fig. 3) device of SensAble, available at the VR laboratory of the "Dipartimento di Meccanica" of Palermo has been used.

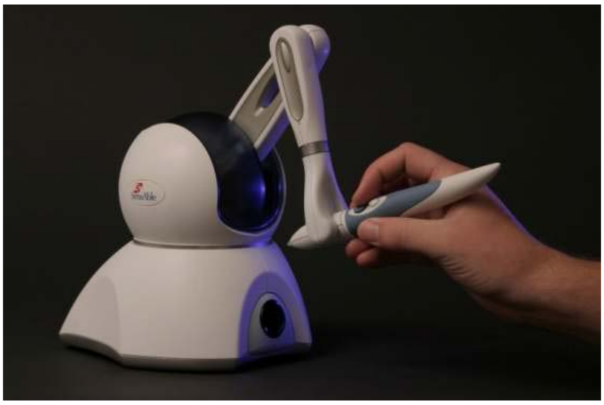


Figure 3 – Phantom Omni haptic device

However some modifications are being produced in this system so to allow its use through one or two fingers, following the scheme of other finger-based haptic systems like that shown in figure 4.

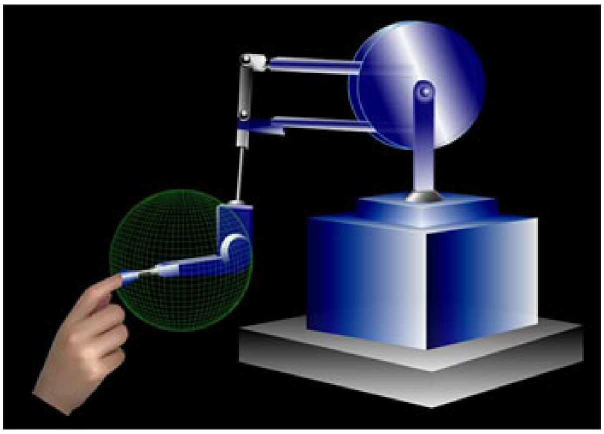


Figure 4 – A finger based haptic device

For what concerns the software characteristics, it has been decided to use an advanced programming language like the C++, in order to allow any improvement and modification of the application in a very fast and easy way.

Two kind of open source libraries have been used: OpenHaptics and OpenGL. The first is related to the haptic rendering while the second to the graphic one.

The graphic rendering is used to setup the scene visualization and manage its changes in real time.

The haptic rendering, instead, is used to define the objects' properties and evaluate the forces following the mutual collisions among objects. The haptic rendering is the most distinctive aspect of a force-feedback application.

4.1. Haptic rendering

Through the haptic rendering the geometric and tactile properties of the objects are defined, the mapping and the management of the real and virtual spaces are performed, and the output forces are evaluated.

The mapping of the spaces establishes a relationship between the working volume, in the real world and in the simulated one, of the haptic device.

With the haptic rendering the shapes of the objects and the related parameters can be defined. These parameters are used to define the tactile properties of the virtual objects, identify which sides of the surfaces must be used to apply the haptic effect, set the stiffness, the kind of damping, etc.

However the prime importance aspect in the haptic rendering is the calculation of the force-feedback. This evaluation is very simple if only rigid bodies are analyzed, while when working with deformable bodies, the basic model of the haptic rendering cannot be used. It is so necessary referring to methods able to calculate the reaction forces following the bodies' deformations. Among these calculation techniques, the most used are based on the:

- Finite Element Method (FEM),
- Boundary Element Method (BEM),
- Mass Spring Model (MSM).

The first two methods give very reliable results but they need very long calculation times that, in general, do not allow their use in real-time application.

The MSM, instead, although provides approximate results, need processing times that, particularly for simple models, are very short and that allows its large use for applications that need real time force-feedback evaluation, like the application here introduced.

4.2. Mass Spring Model

The Mass-Spring Model (MSM) schematizes the surface of a deformable object as a net of masses and springs, as shown in figure 5.

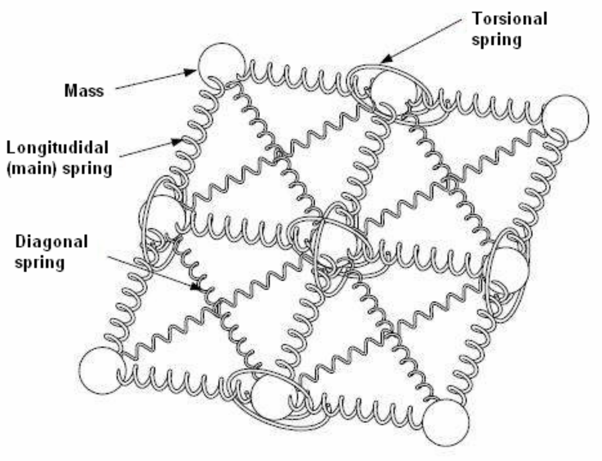


Figure 5 – The MSM scheme

The system parameters like position, speed and acceleration of the masses, are regulated according to the newtonian laws.

Three kinds of springs can be defined (fig. 5): the longitudinal main springs, that keep the distance between adjacent masses, the diagonal springs, that provides the transversal resistance, and the torsional springs that are employed for maintaining the right angle value between the longitudinal springs.

The basic MSM scheme can be also enriched with other objects to simulate dumping forces or other effects.

Both external and internal forces can be applied on every single mass. External forces are those applied by colliding objects whereas internal forces are those exchanged among the masses through the net of joining springs.

The total force F_i applied to the generic mass i is equal to:

$F_i = f_D + f_L + f_T + F_e$;

where:

- f_D is the dumping force;
- f_L represents the force due to the longitudinal and transversal springs;
- f_T is the force due to the torsional springs;
- F_e is the external force.

For the evaluation of the force-feedback it is necessary to calculate the external force.

Almost all the introduced parameters are dependent on the time so, as a consequence, the MSM scheme can be described through a system of differential equations. By giving the initial conditions (like position, speed and acceleration) at the starting of the application, it is possible to solve the system of equations. This can be made through different methods, the most used are the “Eulero” and the “Runge-Kutta” integration techniques.

These methods combine good quality approximate results, little time processing and calculation easiness; all that goes with the setting up of real time applications.

It is very important to note that the mass spring method can be used both for linear and not linear materials, with elastic or plastic behaviour; it only needs giving as input the right parameters’ values .

4.2.1. Evaluation of the external forces

In the application here introduced the only deformable structure that makes use of the MSM is the skin. The surface is meshed into quadrangular polygons of which all the vertexes have a connected mass (fig. 6).

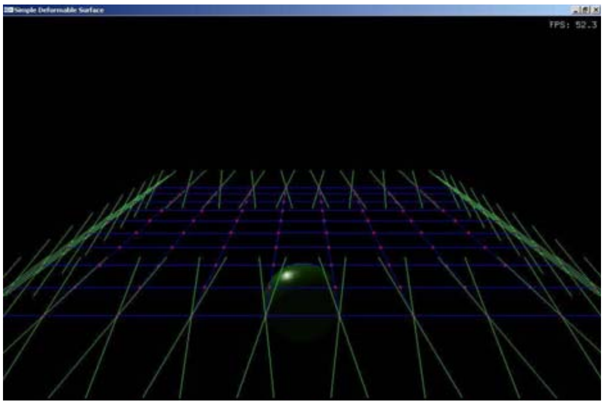


Figure 6 – The adopted MSM model of the skin

Longitudinal springs are placed along the sides of the polygons; moreover, in order to dissipate energy so that allowing the return at the initial position without oscillations around the position of equilibrium, a dumping effect depending on the speed of the masses has been implemented.

The dynamics that regulates the mass spring model can be described through a system of differential equations like this:

$$(\dot{x}_i) = (v_i) \tag{1}$$

$$(\dot{v}_i) = \left({f_i}/{m_i} \right) \tag{2}$$

where x_i e v_i are the position and velocity vectors, while fi is the total force on the mass m_i.

The force fi can also be written by an explicit formulation:

$$f_i = -k_d v_i + K_m \sum_j \frac{l_{ij}}{|l_{ij}|}\left(|l_{ij}| - r_{ij}\right) + f_i^e \tag{3}$$

where k_d is the dumping factor, K_m is the spring stiffness, l_{ij} represents the displacements vector, r_{ij} is the length of the spring and f_i^e the external force.

Equations 1 and 2 can be written in a more compact way:

$$\left[\frac{\dot{x}}{\dot{v}}\right] = \left[\frac{v}{a}\right] = \left[\frac{v}{f/m}\right] \tag{4}.$$

Equation (4) represents a system of first order differential equations depending on the time t. By replacing the explicit formulation of the external force f_i^e (eq.3) in the previous system (eq.4) and giving the initial conditions, it is possible to solve the system and find the value of f_i^e.

This can be made with the Eulero method. By considering a little time variation Δt around the initial time t_0, it is possible to obtain a rounded solution by writing the equations (4) in the following way:

$$\left[\frac{x(t_0+\Delta t)}{v(t_0+\Delta t)}\right] = \left[\frac{x(t_0)+\dot{x}(t_0)\Delta t}{v(t_0)+\dot{v}(t_0)\Delta t}\right] = \left[\frac{x(t_0)}{v(t_0)}\right] + \left[\frac{v(t_0)\Delta t}{f(t_0)\Delta t/m}\right] \tag{5}$$

When the application is running and the surface is deformed, as a consequence of a collision, the lengths of the springs (l_{ij}) and the velocities of the masses change. By introducing the up-to-date springs lengths l_{ij} in the equation 3 and substituting it in the equation 5, the external force f_i^e can be evaluated.

4.3. Surface reconstruction

When the system of differential equations (4) is solved, the position, the speed and the acceleration of the masses are calculated. Thanks to these information is possible to redraw the up-to-date surfaces of the virtual objects deformed as a result of a collision.

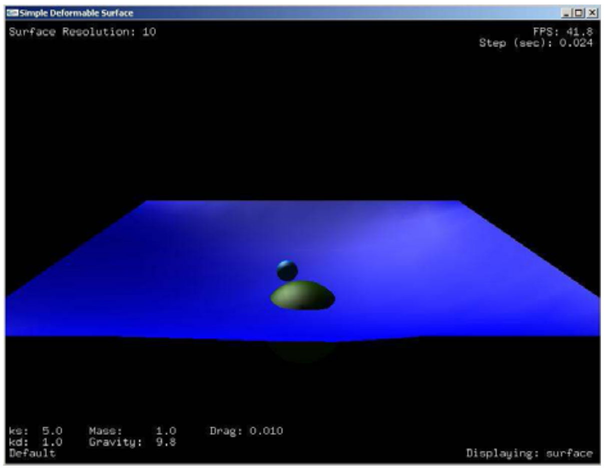
To make that many methodologies are available, among all, one of the most famous is based on an algorithm proposed by Dachille et al. (1999). Such technique suggests to use the knots of the mass spring model as control points of the B-spline geometries. Moreover, by using this method, it is possible to give different weights to the knots that have larger or lower displacements so obtaining a surface that represents in a very precise way the shape of the MSM mesh. This technique turns out to be efficient, intuitive and fast to implement.

The density of the mesh can be related to the resolution of the surface and, by planning different kind of meshes, differences and precisions of the calculation can be verified. It is also possible to work on multi resolution surfaces (Gao et al. 2006) to evaluate local effects.

5. APPLICATION CHARACTERISTICS

Considering the characteristics of the palpation manoeuvre, like for example the possibility to use one or more fingers, or that the contact is only on the external surface of the body, that the movements of the internal organs are constrained, it has been possible to impose some simplifications during the planning of the developed application. One of these is related to the objects to simulate that, in this case, are only three: the fingertip, the skin and the anatomical formation, that have been implemented through three suitable C++ classes.

It is important to notice that all the objects, with the exception of the proxy (that needs only the graphic rendering), must be defined twice: the first one for the haptic rendering (the object is tangible but not visible) and the second one for the graphic rendering (the object is visible but not tangible).



```
Surface Resolution: 10                                   FPS: 41.8
                                                  Step (sec): 0.024

ks:  5.0    Mass:     1.0     Drag: 0.010
kd:  1.0    Gravity:  9.8
Default                                          Displaying: surface
```

Figure 7 – Objects represented in the application

5.1. Finger

The virtual finger used to touch and to interact with the skin has been approximate with a small sphere, the blue one in fig. 7, having suitable dimensions (as regards other objects dimensions).

This object, also called “proxy”, is analogous the mouse pointer of any window-based software application, and is used to interact with other virtual objects.

This kind of scheme has been adopted since only the fingertip is in contact with the skin.

Kinematics and dynamics characteristics of the object "fingertip", like the working volume, the maximum measurable speed, etc.., are related to the characteristics of the haptic peripheral.

Since the interaction among different deformable objects (Ruan et to the. 2004) is generally more complex to manage in regard to the force-feedback calculation, it is convenient, if possible, to use rigid bodies characteristics for the virtual objects.

In this case, even if the fingertip can have elastic deformations, these usually are very little and so negligible. For this reason the proxy (fingertip) has been considered as a rigid body.

Other main properties of the proxy are the following:

- it must be able to push the surface of the skin but not to drag it;
- the push must be applied only on a side (the external one) of the skin, so the proxy cannot go beyond it;
- the interactions with the anatomical formation must be indirect, in fact the tactile perception must happen through the skin.

5.2. Anatomical formation

The second object is the anatomical formation, representing a nodule, that has been represented as a bigger sphere in comparison with the proxy (fig.7).

In this case the graphic rendering is not so important like the haptic rendering, that is instead essential.

The contact with other objects can take place only through its external surface. This object, like the fingertip, has also been modelled as non-deformable body. In fact, since the contact between the finger and the skin happens on a reduced surface and with a very low external load, the tissue between the skin and the anatomical formation, distributing the external load on a wide surface, stresses the nodule in a very light way, causing very small elastic deformations. For this reason it is reasonable to hypothesize its behaviour similar a rigid body.

5.3. Skin

The last object to simulate is the skin. It has been modelled as a planar and deformable surface. An “integrated” MSM system has been used for its haptic and graphic representation. The system can be defined “integrated” because the same points’ coordinates have been used to define the MSM mesh and the vertexes of the polygons defining the surface. Certainly in this way, the graphic rendering accurately follows the haptic rendering.

As regards the haptic rendering, the mechanical characteristics of the tissue have been defined by setting the parameters of the MSM system.

Other characteristics that do not depend on the MSM system, like the rules that regulate the interaction and the collision with other objects, have been defined through the implementation of dedicated classes in C++ language.

5.3.1. Tissue mechanical characteristics

A large part of the soft tissues is generally constituted of collagen, elastin and polysaccharides.

Collagen is a protein with a fibrous structure and has the purpose to limit the deformations and prevent the mechanical breaking of the tissue. The fibres have a helical shape then, if submitted to an axial load, initially show a very low “shape” stiffness, once the fibres become straight, instead, the mechanical properties considerably increase, because of the very high strength of the intra and intermolecular bonds.

Elastin is also a structural protein with high elasticity values but generally lower than the collagen.

Polysaccharides are polymers made of sugars and have viscous characteristics.

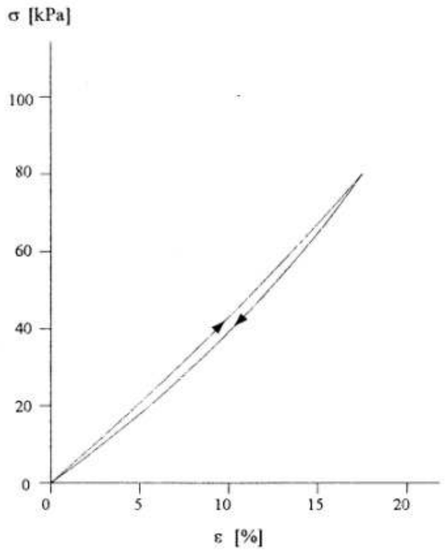


Figure 8 – Mechanical characteristics of a tissue

The skin tissue so has an elastic not linear behaviour. Moreover the loading and unloading curves slightly differ, describing a little hysteresis loop as shown in figure 8.

In the presented application, considering the typology, the values of the applied loads (at the most some dozens of grams) and the thickness of skin (about 1-2 mm), the tissue has been modelled like a perfectly elastic material , without making a big error.

5.4. Algorithm for collision detection

The available algorithms in the OpenHaptics library allow to evaluate only the collisions between the proxy and other objects, but do not allow to detect any collision between other user–defined objects. For such reason the OpenHaptics algorithm has been used to evaluate the collision between the proxy and the skin, whereas another technique has been used for the collision detection between the sphere, representing the nodule, and the surface, representing the skin. This has been necessary because there can be no interpenetration between the skin and the anatomical formation.

It has been used an algorithm based on the implicit formulation of the surfaces that, in this particular case, has been implemented in a simplified way because one of the two objects (the sphere representing the nodule) is fixed.

Moreover another simplification has been performed in analysing the collision between the proxy and the nodule rather than evaluating the one between the skin and the nodule. That has been possible because, even if the real collision is between the skin and the nodule, in effect the skin is moved by the proxy and it always has its same positions. For this reason, to simplify the procedure, in the following all the calculations, even if related to the proxy, are equally valid for the skin.

The implicit formulation of a sphere is the following:

$$(x-x_0)^2+(y-y_0)^2+(z-z_0)^2-{r_n}^2=0 \qquad (6)$$

where x_0, y_0, z_0 are the coordinates of the centre and r_n is the radius of the sphere.

In particular for the sphere representing the nodule, the used values are the following:

$$\begin{cases} x_0=0 \\ y_0=2 \\ z_0=0 \\ r_n=1 \end{cases},$$

so the equation 6 can also be written in this form:

$$f(x,y,z)=x^2+y^2+z^2+4y+3 \qquad (7).$$

By changing the equation 7 into an inequality it is possible to select the internal points ($f(x,y,z)<0$) and the external ones or those on the sphere surface ($f(x,y,z)\geq 0$).

Because the coordinates of the proxy are always known, it turns out very simple to evaluate if there is any collision. If so the proxy is inside the sphere and it is necessary to drag it on the surface.

By supposing the proxy is just a little inside the sphere, it is possible to calculate the point on the surface where the proxy must be moved, by dragging it along the direction joining the proxy position and the centre of the sphere. It needs then to calculate the straight line passing through the proxy (which has coordinates P_x, P_y, P_z) and the centre of the sphere; the parametric formulation of this line is the following:

$$\begin{cases} x = P_x * t \\ y = -2 + \left(P_y + 2\right) * t \\ z = P_z * t \end{cases} \qquad (8).$$

By replacing the equation 8 in the equation 6 it is possible to calculate the value of the parameter *t* that verifies the system of equations 8 and 6:

$$t_1 = \sqrt{\frac{1}{\left[P_x^2 + (P+2)^2 + P_z^2\right]}} \qquad (9).$$

Using this value (t_1) in the equation 6, the coordinates of the intersection point *S* between the straight line and the sphere are known.

This procedure would be correct if the proxy is a point. Because it has a spherical geometry, the previous calculation must be lightly modified taking into account the radius r_p of the sphere representing the proxy.

Repeating the previous described procedure, it is possible to find the right value of *t* :

$$t_2 = \frac{r_p}{\sqrt{P_x^2 + (P+2)^2 + P_z^2}} + t_1 \qquad (10)$$

By using this values of *t* in the equation 6, the point on which moving the centre of the proxy after the collision with the anatomical formation can be evaluated.

6. USERS’ TEST AND FUTURE WORKS

The application has been tested by some users, some of these are doctors other, instead, engineers. All of them have shown a good familiarity with the haptic device and the system, and have been able, in a simple and fast way, to identify the subcutaneous nodule, of which they have understood the shape, the dimensions and the consistency.

Some of them have complained about the absence of other internal organs and about the shape of the surface used to simulate the skin. This last, in fact, has been simulated with a not much intuitive and realistic planar quadrangular surface.

For these reasons most of the future works will be addressed to the modelling of a larger portion of skin and the use of curved surface to simulate in a more realistic way the external shape of the human body.

Moreover other internal organs and some parts of the bony structure will be added in the application.

It will be also necessary to modify opportunely the MSM method to apply it to volumes and so being able to simulate also the mechanical properties of the internal organs, modelled as deformable bodies.

It will be also necessary to modify the collision detection algorithm because it can be used only for geometric shapes that can be represented through explicit and continuous functions, and that represents a particular working condition.

To simulate in a more realistic way the mechanical properties of the skin, it will be studied the use of more complex multilayer and multimaterial physical models (Yang et al. 2004).

Finally, it could be interesting to study the integration of the application in a virtual reality environment in order to give the users the possibility of a more realistic graphical perception of the objects.

7. CONCLUSIONS

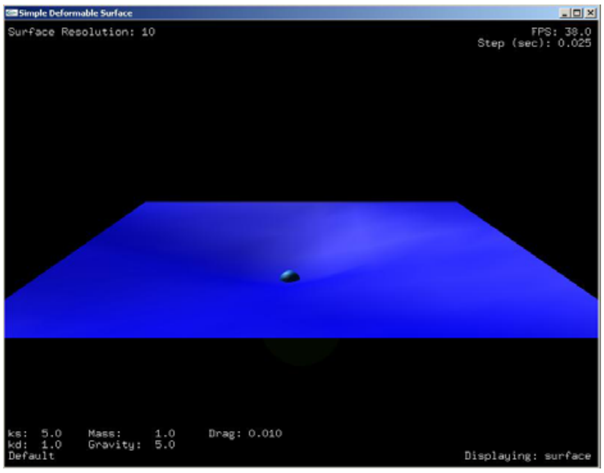
The innovative aspects and the advantages introduced by the haptic technology are, without any doubt, remarkable and will certainly increase more and more with the improvements of the working techniques and devices.

In this context, the research activity here presented, represents a promising innovative methodology for the medical palpation training.

Thanks the presented system an user can “touch” a virtual representation of the human skin a perceiving it as real. It is possible, in fact, thanks the use of the developed force-feedback system (fig. 9) to feel the contact with a surface, representing the skin and, through a palpation manoeuvre, understand if there is some problem, like the presence of a nodule, a particular anatomical formation.

Although the system has been implemented in a simplified way (fig. 9), all the users that have tested it, think the proposed technique can represent a very powerful tool for the training of not skilled doctors.

The problem of the force-feedback calculation has been solved making use of the MSM and a simple but very efficient algorithm has been implemented for the collision detection among the rendered objects.



```
Surface Resolution: 10                    FPS: 38.0
                                Step (sec): 0.025

ks:  5.0    Mass:     1.0    Drag: 0.010
kd:  1.0    Gravity:  5.0
Default                           Displaying: surface
```

Figure 9 – a screenshot of the developed application

In general it can be stated that the obtained results are very promising. In the development of similar future applications, the proposed methodology can represent a valid reference to analyze more complex conditions and to model other objects with different dynamics and structures.

ACKNOWLEDGMENTS

Authors want to remember sincerely Prof. Francesco Cappello, scientist, teacher and friend recently dead.

REFERENCES

Basdogan C., Ho C., Srinivasan M.A., 2001. Virtual Environments for Medical Training: Graphical and Haptic Simulation of Common Bile Duct Exploration. *IEEE/ASME Transactions on Mechatronics*, 6(3):267-285

Bouzit M., Popescu G., Burdea G., Boian R., 2002. The Rutgers Master II-ND Force Feedback Glove. *Proceedings of IEEE VR 2002 Haptics Symposium*, March 24-27, Orlando FL (USA)

Burdea G., Patounakis G., Popescu V., Weiss R.E., 1999. Virtual Reality-Based Training for the Diagnosis of Prostate Cancer. *IEEE Transactions on Biomedical Engineering*, 46(10): 1253-60

Dachille, F., Qin, H., Kaufman, A. and El-Sana, J., 1999. Haptic Sculpting of Dynamic Surfaces. *Proceedings of Symposium on Interactive 3D Graphics,* pp.103-110, April 26-28, Atlanta, Georgia (USA)

Dang T., Annaswamy T., Srinivas M.A., 2001. Development and Evaluation of an Epidural Injection Simulator with Force Feedback for Medical Training. *Medicine Meets Virtual Reality,* Amsterdam, IOSPress

Delingette H., 1998. Toward Realistic Soft-Tissue Modeling in Medical Simulation. *Proceedings of the IEEE*, pp. 512-523, April 21-23, Adelaide, South Australia

Gao Z., Gibson I., 2006. Haptic sculpting of multi-resolution B-spline surfaces with shaped tool. *Computer-Aided Design,* 38:661–676

Harris E., Morgenthaler G., 2004. Planning, implementation and optimization of future space missions using an immersive visualization environment (IVE) machine. *Acta Astronautica*, 55: 69 – 78

Heinrichs W.L., Srivastava S., Brown J., Latombe J.-C., Montgomery K., Temkin B., Dev P., 2000. A Steroscopic Palpable and Deformable Model: Lucy 2.5. *Proceedings of Third Visible Human Conference*, October 5-6, Bethesda, Maryland (USA)

Langrana N., 1997. Human Performance Using Virtual Reality Tumor Palpation Simulation. *Computers & Graphics*, 21(4):451-458.

Liu X., Dodds G., McCartney J., Hinds B.K., 2004. Virtual DesignWorks-designing 3D CAD models via haptic interaction. *Computer-Aided Design,* 36:1129–1140

Ma W., Zhong Y., Tso S., Zhou T., 2004. A hierarchically structured and constraint-based data model for intuitive and precise solid modeling in a virtual reality environment*.* *Computer-Aided Design,* 36:903-928

Montgomery K., Bruyns C., Brown J., Sorkin S., Mazella F., Thonier G., Tellier A., Lerman B., Menon A., 2002. Spring: a general framework for collaborative, real-time surgical simulation. *Studies in health technology and informatics,* 85:296-303

Okamura A.M., 2004. Methods for haptic feedback in teleoperated robot-assisted surgery. *Industrial Robot,* 31:499-508

Ruan H.H., Yu T.X., 2004. Collision between mass–spring systems. *Impact Engineering,*31:267-288

Tendick F., Downes M., Goktekin T., Cavusoglu M.C., Feygin D., Wu X., Eyal R., Hegarty M., Way L.W., 2000. A Virtual Environment Tested for Training Laparoscopic Surgical Skills. *Presence*, 9(3): 236-255

Wang P., Becker A.A., Jones I.A., Glover A.T., Benford S.D., Greenhalgh C.M., Vloeberghs M., 2007. Virtual reality simulation of surgery with haptic feedback based on the boundary element method. *Computers and Structure,* 85:331-339

Yang Z., Chen Y., 2005. A reverse engineering method based on haptic volume removing. *Computer-Aided Design,* 37:45-54

AUTHORS BIOGRAPHY

Tommaso Ingrassia is a research assistant at the “Dipartimento di Meccanica dell’Università di Palermo”. He received his PhD in Mechanical Engineering in 2007. He was a Visiting Researcher at the Graphitech Foundation. He is responsible of the ‘Virtual Reality Lab’ of the University of Palermo. His main research activities are about redesign methodologies, virtual reality in mechanical design, haptic systems, composite material, multiobjective and topological optimisation.

Vincenzo Nigrelli is a Full Professor of Machine Drawing at the “Dipartimento di Meccanica dell’Università di Palermo”. He is author of more than 100 papers. His present activity is: reverse engineering, haptic systems, topological and shape optimization, image processing, and non-destructive evaluation. He is active in national research programs (PRIN), assignor of research funds, team leader of EU project under the 6th Framework Research Programme. He teaches Fundamentals of Industrial Design and Computer Aided Draft.

A WEB3D APPLICATION FOR THE BIN-PACKING PROBLEM

Bruno Fabio[(a)], Caruso Francesco[(b)], Pisacane Ornella[(c)],
Musmanno Roberto[(d)], Muzzupappa Maurizio[(e)], Rende Francesco[(f)], Giovanni Venuto [(g)]

[(a) (b) (e)]Università della Calabria – Department of Mechanical Engineering – Rende (CS) - Italy
[(c) (d) (f) (g)] Università della Calabria – Department of Electronics, Informatics and Systems – Rende (CS) - Italy

[(a)]f.bruno@unical.it, [(b)]francesco.caruso@unical.it, [(c)]opisacane@deis.unical.it,
[(d)]musmanno@unical.it , [(d)]muzzupappa@unical.it , [(d)]frende@itkey.it

ABSTRACT

This paper describes a web application for supporting the user in the assessment of the optimal loading configuration for several carriers (i.e. trucks, containers, ship, etc.). The application has a standard form-based user-interface to insert data of the available carriers and items have to be loaded. The solution of this problem (known as Bin-Packing Problem (BPP)) is found by a specific algorithm and visualized by a 3D graphics representation inside the web page. Although some commercial applications already exist, ours runs directly on the web and offers, at the same time, an efficient and robust solver and a 3D visualization allowing the user to better understand the localization of the items inside the carrier and to interactively change some of the problem constraints directly on the 3D representation.

Keywords: Web3D, logistics, bin-packing

1. INTRODUCTION

The BPP is one of the most relevant optimization problem in logistic and transportation field and represents a classical operative problem for better managing the distribution centres. The main goal is to optimize the carriers utilization, i.e. to minimize the total number of used carriers satisfying some constraints. The common constraints, presented in literature, are on the carriers capacity. Moreover, in some realistic scenarios, other technological constraints are imposed too: load stability, items priority, items overlapping and rotation. From a mathematical point of view, the BPPs are NP-Hard and so the exact algorithms are not used for solving them. On the other hand, the heuristic methods are not able, in some cases, for giving real time solutions because it is very difficult to formalize all the constraints. In fact some relaxations, exact and approximate algorithms have been studied and introduced by Gilmore & Gomory (1965) and by Johnson (1973).

In a general framework, two main entities can be distinguished: the items to be loaded and the transportation carriers, called bins. A bin could be indifferently a vehicle or a container/pallet. The basic unit is not defined by the item but by the wrapping, i.e. the envelope containing the items. It is done for assuring a more accurate transportation of the items even if it causes some additional costs and weight increments. The interpretation of the BPP results is usually a complex task because it represents the spatial collocation of each item in the bins. Moreover it is often necessary to interact with the solution in order to change some constraint or to retrieve specific data as the item properties. Since our scope of interest is the tri-dimensional BPP, it has been considered to use 3D Graphics as the development tool able to present the results in an intuitive way, allowing the user to query the system and to interactively change any parameters.

3D Graphics is becoming a common tool for the development of advanced user-interface in several application fields. In logistics, for example, there is a wide range of researches that put in evidence the advantages that may be obtained using 3D to support simulation or optimization software. Weznel (2003), proposes a taxonomy of visualization techniques for simulation in production and logistics. He outlines how to use it as a base for decision support to select the right visualization technique for specific target groups.

Bruzzone and Giribone (1996) present one of the first visualization tool that uses together web technologies and 3D graphics as a wired decision support system for logistics.

As regards commercial software for solving the BPPs, two of the most complete ones are Cube-IQ and CubeMaster.

The former gives optimal volume/weight utilization, using its own database, data import/export and reporting. After finding the final configuration, it produces clear 3D load diagrams for seeing it. It also offers full flexibility in loading and stacking rules, which can be defined for each orientation of a box.

CubeMaster optimizes the cargo load on trucks, air & sea containers and pallets quickly and efficiently and supports in planning the order picking, loading and capacity requirements. It allows users to compute the optimal loading diagram and layout, to forecast the best size of orders, to maximize the loading space and weight in cube, and finally to build quickly the reports in 3D graphics.

2. THE BIN PACKING PROBLEM

The mathematical formulations of the BPPs can be defined as linear problems with binary decision variables. In the bi-dimensional problems both items and bins are defined by two parameters individuating a surface. Instead, in the tri-dimensional problems the items have a very low density and so the bins capacity is expressed in term of their volume rather then their weight. For both these problems, we assume a regular shape of the items (also cylindrical or/and composite shapes) and items disposition parallel or orthogonal respect to the bin axis. The items have been classified into three categories for taking into account overlapping constraints: normal, lightweight and heavy items. The bins have been considered with a regular shape and an utilization cost and a capacity are associated to each of them. Our solver considers only bi and tri dimensional bin packing problems under constraints on:

- Overlapping: it is possible to overlap items taking into account their nature (normal, lightweight and heavy) and each item has a maximum weight toleration;
- Rotation: it is possible to rotate the items when specified;
- Priority: when the items have multiple destinations, it could be very important to respect a precise loading sequence (for example items for the same destination have to be loaded near to each other thanks to an interaction among the loading and the itinerary choice (vehicle routing problems)).

Both problems are solved by an approach called Tabu Search Pack (TSPack), i.e. a meta-heuristic that iteratively tries to improve the quality of the current solution (Lodi, Martello and Vigo 2004). This approach, in both cases, has been modified for taking into account new constraints like items rotation and their overlapping. The Tabu Search algorithm starts with an initial solution, obtained considering a total number of bins equal to the number of items and assigning each item to a specific bin, and tries to improve the current solution by applying some feasible moves on the base of the specific problem instance.

2.1. The Tabu Serach Algorithm

The Tabu Search Pack (TSpack) is a meta-heuristic applied for improving in an iteratively way the current solution until the optimal configuration is obtained or a maximum number of iterations is performed (Lodi, Martello and Vigo 2004). It is assumed to have n items with a width wj , height hj and depth d_j ($j \in J = \{1,\ldots n\}$)), and an unlimited and identical number of bins with width W, height H, and depth D. The priority and overlapping constraints are considered in the internal heuristic A producing a feasible solution for a given problem instance. Let be L a lower bound on the optimal solution and z* the cost of a feasible solution, if z* = L the procedure ends; otherwise the search starts generating an initial solution loading each item in a different bin. Some tests have shown that the objective functions of the BPPs are flat, i.e. many solutions using the same number of bins exist. TSpack introduces a characterization of equivalent solutions based on the fact that a less used bin (target bin) exists and so it is easier to empty it. The target bin is the one that minimizes the filling function :

$$\varphi(Si)=\alpha\ (\Sigma\, j \in Si\ vj) \backslash V - Si \backslash n ; \tag{1}$$

where $vj = wj * hj * dj$ and $V = W * H * D$ for 3BP; instead $vj = wj * hj$ and $V = W * H$ for 2BP. Si is the set of items loaded in the bin i. When the target bin is found, the algorithm tries, through some moves, to empty it changing the loading of a sub-set S of items. S is made up by an item j coming from the target bin plus the ones in the other k bins, where k defines the current neighbourhood. The current configuration is obtained applying the algorithm A to the set S. The important parameters of TSpack are three: K, maximum number of iterations (NitMax) and time limit (TL). The two last ones quantify the algorithm duration (i.e. the number of iterations performed and the total computational time). It means that the TSpack procedure is stopped even if it has not found the optimal solution after a fixed number of iterations or when a time limit is reached. These two parameters are constant during the search and they have to be opportunely initialized (i.e. they should be directly proportional to the number of items n to be loaded). If n is large enough and if the NitMax and/or TL values are not proportionally large, the algorithm could not reach the optimal solution. On the other hand if these values are increased too much, the computational time could be exponential. The most important parameter is K, i.e. the search breadth, and it is independent from the instance and varies dynamically during the algorithm.

The moves applied during the procedure try to empty the target bin changing the loading configuration of the set S. The new solution cost is obtained summing the number of the bins produced by the internal heuristic A applied to the set S and the remaining bins that contains the $\{1,\ldots n\} \backslash S$ items. The solutions obtained are considered feasible if the loading configuration is changed and the total number of bins is not increased respect to the current solution. In other words, the moves try to remove the item j from the target bin without using a new bin.

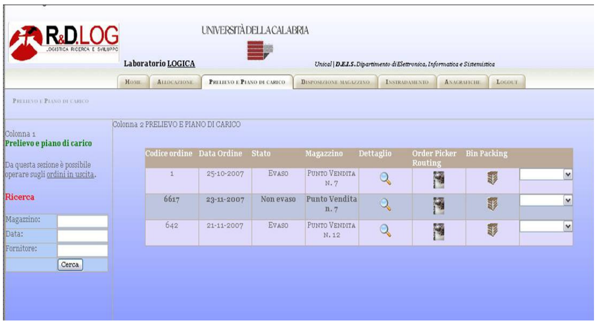
The internal heuristics A is applied for each item j loaded in the target bin and for each subset of k bins. The value of K is increased by one if the search does not find feasible moves.

If the search finds either a feasible move that decreases the total number of bins or a non-tabu move that loads the set S of items within k bins, it is immediately applied and the value of k (i.e. the dimension of the subset of bins considered in the moves, initialized to 1) is decreased by one. If a move is not immediately applied then a penalty is computed whose value is infinite either if the move is tabu or the internal heuristics uses at least two extra bins. The move with minimum penalty is chosen if the search does not find any to be immediately applied. When K reaches a maximum value, a diversification procedure is applied.

About the internal heuristic, both for 2BPP and 3BPP, Hybrid Next Fit (HNF) algorithm is considered adapting it for taking into account the capacity constraints for the bins and the priorities, the overlapping and the rotation constraints for the items. After some adjustments, it becomes Hybrid HNF (HHNF). In the 3D case, for example, the routable items are positioned firstly with their shorter side as height in a way that they can occupy the minimum vertical space. Then they are horizontally rotated with the longer side as width. The items are firstly sorted by their specific class (heavy, normal and lightweight) and then by their priority. The HHNF is a section-level-procedure. A section has width equal to the one of the bin and length equal to the longest item inside. A level has height equal to the one of the highest item inside. Then the current item is loaded in the upper left side of the current section if possible; otherwise the section is closed and another is open. If it is not possible, then the item is loaded in the next level. It means that the current level is closed and another is open. If in the bin there is not sufficiently space for loading the item, it is closed and another is open.

3. FRAMEWORK OVERVIEW

The proposed framework it is made up by three main software entities, which can run on separate machines. These entities are the BPP solver, described in the previous section, the Web3D application, and the web interface to the global framework. This latter is a simple JSP page, in which the user can select the order he wants to process, as showed in Figure 1.


R&D.LOG
LOGISTICA RICERCA E SVILUPPO
UNIVERSITÀ DELLA CALABRIA
Laboratorio LOGICA
Unical | D.E.I.S. Dipartimento di Elettronica, Informatica e Sistemistica
HOME
ALLOCAZIONE
PRELIEVO E PIANO DI CARICO
DISPOSIZIONE MAGAZZINO
INSTRADAMENTO
ANAGRAFICHE
LOGOUT
PRELIEVO E PIANO DI CARICO
Cerca

Colonna 1
Prelievo e piano di carico

Da questa sezione è possibile operare sugli ordini in uscita.

Ricerca

Magazzino:
Data:
Fornitore:

Colonna 2 PRELIEVO E PIANO DI CARICO

Codice ordine	Data Ordine	Stato	Magazzino	Dettaglio	Order Picker Routing	Bin Packing	
1	25-10-2007	EVASO	PUNTO VENDITA N. 7				
6617	23-11-2007	Non evaso	Punto Vendita n. 7				
642	21-11-2007	EVASO	PUNTO VENDITA N. 12				

Figure 1: The JSP page for the order choice

All the orders information in fact, are stored in a database, and for each order there is the list of the items. The user specifies via web page the order he wants process. Afterwards, the JSP web page retrieve the order information from the database, and send them to the BPP solver. This latter computes the BPP solution and then produces the solution file. Subsequently, the JSP page returns a new html page made up by an applet and a web3d plug-in. Finally, the applet loads the optimized solution previously generated by the solver and stored as file, and dialogues with the visualisation plug-in in order to recreate the 3d scene in accord with the bin-packing solution. The visualisation plug-in receives the information relative to all the items and the carriers, and then "translate" them in a visual output. This dialogue between the applet and the visualisation plug-in occurs through an Inter-Process Communication (IPC) channel developed for this purpose. All the details about the communication are explained in the next section. In the Figure 2 is represented the temporal diagram of the actions of the different software entities.

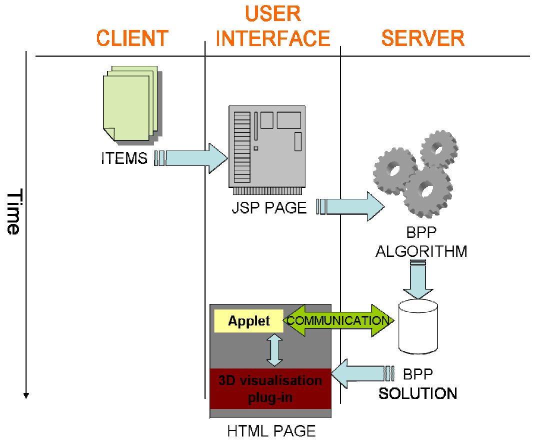


Figure 2: Temporal diagram of the actions of the software entities

The Web3D application has been developed in order to have an easy and flexible interface to interact with the BPP solver, implemented on the web server side. The application offers several functionalities:

1. The optimal BPP solution visualization in 3D graphics.
2. The camera movement to observe the solution from different point of view.
3. The data retrieval about the properties of each item.
4. The movement of the items placed in the optimal solution and the querying of the BPP solver for the feasibility check.
5. The collisions detection during the manual movements of the items.

The final result is the three-dimensional visualisation of the optimal BPP solution for a specific order, as shown in Figure 3. Further, the high level visualisation environment is able to perform advanced visualisation technique not available in the other software. It is possible, e.g., to set the transparency of a 3D entity, like showed in Figure 4.

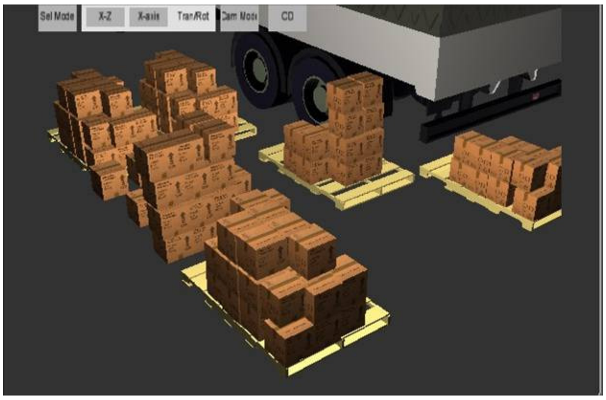


Figure 3**:** 3D visualization of the solution

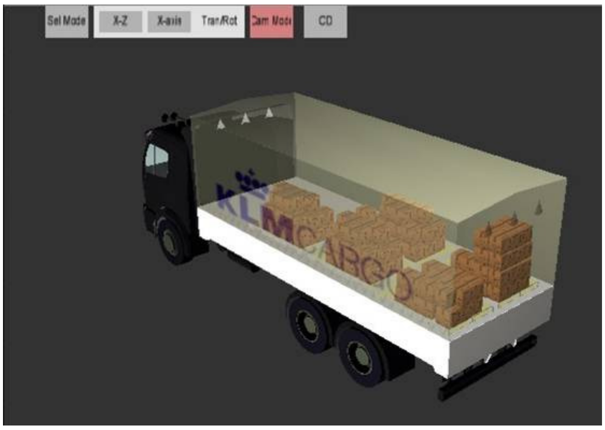


Figure 4**:** Visualization of items within a transparent carrier

4. SOFTWARE ARCHITECTURE OF THE WEB3D APPLICATION

The Web3D application has been implemented using a commercial software named Virtools Dev (www.virtools.com). It is a complete development and deployment platform for the creation of interactive three-dimensional contents. Using Virtools in fact, it is possible to create a virtual environment by importing 3D objects in several file formats. Further, it is possible to define a behaviour for each object. Virtools, in fact, has its behavioural engine, and it is able to process user events and to execute the behaviour in accord with them. Further, it has its own Software Development Kit (SDK), and therefore it is possible to extend and customize Virtools in order to create new functionalities. As regards our goal, there was the need to interface Virtools with a Bin-Packing solver. In our aims, in fact, the Bin-Packing solver sets the position of the packs within the carriers, and then the visualization of the solution occurs. The communication between BPP solver and the visualisation environment occurs through an inter process communication channel based on network sockets. The final result is a bi-directional communication channel, which allows to set position and orientation of each 3D model parts (both items and bins) in keeping with the optimal solution, and further, to send input data generated by the user to the solver, like the selection or the movement of an item.

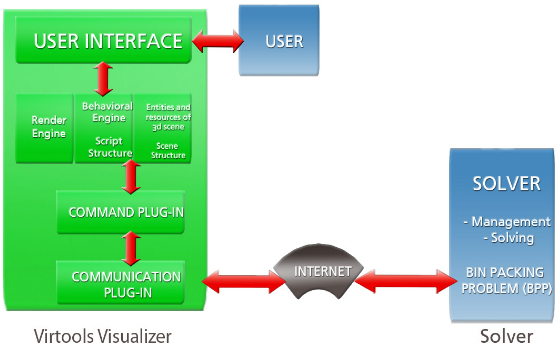


Figure 5**:** Web3D application architecture

The communication channel is based on a new communication library developed ad hoc, the SimLIB library, which will be described in the next section.

4.1. The SIMLib Library

One of the most common way to achieve the Inter Process Communication (IPC) is using TCP/IP sockets.

However, sockets are a low-level communication channel, and they are not easy to use for the transmission of complex and large blocks of data. Therefore, a new library to simplify the data transmission has been implemented. The new library meets the following needs:

1. Simplification of the use of sockets, since many operations are embodied in macro-functions;
2. Typification of transmitted data: it is possible in fact send data in a structured format;
3. Client-server synchronization: some techniques for synchronization, useful for the application context, have been implemented

First of all, a new data type, embodying all the information to be sent, has been defined, as described in Table **1**

Field Name	**Type**	**Description**
Operation	Integer	Create, Delete or modify a pack entry
Id	Integer	Id of the object on affected by the operation
Geometry	Integer	Geometry of the entry (0=cube,1=Cylinder, etc. etc.)
Parent	Integer	Id of the parent of the entry
Color	Float	Color in RGB format

	array	
Position	Float array	Position (X,Y,Z)
Orientation	Float array	Orientation in Eulerian angles
Scale	Float array	Scale factor for X,Y,Z dimension.
Description	String	A description of the entry

Table 1: The SimLib_Data data-type

These fields represent the data to be sent in this specific context, i.e. the BPP visualisation. The SimLib_data can be different for another application of the library.
The functions of the library are described in Table **2:** SIMLib library functions:

Function Name	Description
SIMServerOpen	Create the server end of a communication channel. This end is waiting for a connection
SIMClientOpen	Create the client end of the communication channel. This end connects to the server.
SimLib_synchroSend	Sends a SimLib_data block through the channel
SimLib_synchroReceive	Receive a SimLib_data block through the client

Table 2: SIMLib library functions

With the help of these functions, it is quite easy sending and receiving SimLib_data block of data through a communication channel. The ends of this communication channel are a SimLib_Client and a SimLib_Server.

4.2. Bi-Directional communication between Virtools Dev and the BPP solver

As previously mentioned, it is possible to assign a behaviour for each 3D entity within the 3D visualisation environment. Behaviours are defined by a set of building blocks. Each building block is responsible of a function. Further it is possible to develop a building block using the Virtools SDK. In particular, three new building blocks have been developed in order to achieve our target:

- Master Control Block: It creates a SimLib_Client for the communication and receives data sent by the 3D BPP Solver
- Action Block: It receives the SimLib_data block from the MasterControlBlock and, afterwards, it executes a set of operations in accord with data contained in the SimLib_data block.
- Feedback Block: It creates a new SimLib_Client to send data to the BPP Solver. As previously mentioned, in fact, the user can modify the solution provided by the solver within the 3D environment in a visual way. The configuration hinted by the user is sent to the BPP solver for the feasibility check through the feedback channel, and afterwards the solver sends the

The final 3D composition created with Virtools, made up by three-dimensional entities endowed with a behaviour, can be exported as Web-Page. In this way it is possible to create complex Web3d applications in a simple and efficient way.

5. CONCLUSIONS

The implemented application is a prototype that shows the advantages of using Web3D technologies as an intuitive user interface for a BPP solver.

The main innovation of our application is the web technologies use for remotely solving a common logistics problem. In this way it is possible to solve logistic problems by using a client machine and querying the solver through the network. Further, the solution is presented in a very effective way using a powerful three-dimensional visualisation environment. Moreover the user can modify the optimal solution directly on the 3D visualisation and check the feasibility and the cost of the new solution.

ACKNOWLEDGMENTS

The research presented in this paper has been realized in the context of the LOGICA project (Regional Technological Laboratory of Logistics) co-funded by Regione Calabria, within the POR 2000-2006 Misura 3.16.

REFERENCES

Bruzzone A. G., Giribone P., 1998. Decision-support systems and simulation for logistics: Moving forward for a distributed, real-time, interactive simulation environment. *Proceedings of the IEEE 31st Annual Simulation Symposium,* pp. 17-24. April 5-9, Boston, MA, USA.

Wenzel S., Bernhard J., Jessen U., 2003. A taxonomy of visualization techniques for simulation in production and logistics, *Proceedings of the 35th Winter Simulation Conference: Driving Innovation.* December 7-10, New Orleans, Louisiana, USA.

Gilmore P.C., Gomory R.E., 1965. Multistage cutting problems of two and more dimensions. Operations Research vol.13 pp. 94-119.

Johnson D.S., 1973. *Near-optimal bin packing algorithms*. Thesis (PhD), MIT, Cambridge, MA, USA.

Lodi A., Martello S., Vigo D., 2004. TSpack: A Unified Tabu Search Code for Multi-Dimensional Bin Packing Problems. *Annals of Operations Research*, vol. 131, pp. 203-213.

AUTHORS BIOGRAPHY

Fabio Bruno is an Assistant Professor at the Department of Mechanical Engineering of the University of Calabria. In 2005 he received a PhD Degree in Mechanical Engineering at University of Calabria. He received a Master's degree in Industrial Engineering in 2001. The research activity mainly concerns the application of Virtual and Augmented Reality (VR & AR) to the design process of the industrial products.

Francesco Caruso is a post-doc student at the Department of Mechanical Engineering of the University of Calabria. From 2004 to 2007 he attended his PhD in Mechanical Engineering. His research topics regards Virtual and Augmented Reality, interaction techniques with virtual prototypes and behavioral simulations of virtual prototypes.

Ornella Pisacane is a post-doc student at the Università della Calabria (Cosenza, Italy). Her PhD thesis was concerning with the definition of a simulation-based optimization algorithm for solving the agent scheduling problem in a multi skill call center, supervised by Prof. P. L'Ecuyer (Universitè de Montrèal, Canada) and Prof. R. Musmanno (Università della Calabria). Her current research theme is related to the definition of simulation-based optimization algorithms for efficiently managing supply chains, supervised by Prof. J.F. Cordeau (Universitè de Montrèal, Canada) and Prof. R. Musmanno. She is also working on web services implementation for workflow optimization and for efficiently supporting decision makers related to health care management.

Maurizio Muzzupappa is an Associate professor of Computer Aided Design at the Department of Mechanical Engineering of the Faculty of Engineering of the University of Calabria. From 1989 to 1992, he frequented his Ph.D. at the Department of Mechanical Engineering of the University of Pisa. His current research activities include concurrent engineering (specific topics are the computer support of cooperating virtual engineering teams for design review), collaborative design, virtual and augmented reality and reverse engineering.

Giovanni Venuto was born in Novara (NO), Italy, in 1975. He received his Master Degree in Computer Engineering in 2007. His thesis focused on computer graphics applied to logistic problems. Actually he works in Pitagora s.p.a. as Software Engineer.

Francesco Rende received his Master Degree in Computer Science at the University of Torino. Then he attended the PhD in Operation Research focused on logistics.

Roberto Musmanno is Full Professor of Operations Research at the Università della Calabria (Unical), Italy. His major research interests are in logistics, network optimization and parallel computing. He has published more than 30 papers in a variety of journals and is coauthor of the book: G. Ghiani, G. Laporte, R. Musmanno, "Introduction to Logistics Systems Planning and Control", Wiley, 2004. He is member of the Scientific Committee of the Italian Center of Excellence for High Performance Computing, funded by the Italian Ministry of University. He is member of the editorial board of the international journal Computers & Operations Research (Pergamon) and is chairman of the Council for the Degree in Management Engineering at Unical. He has been involved in several international research projects and was member of the Organizing Committee of international workshops in the field of high performance computing.

MIXED-REALITY ENVIRONMENT BASED ON HAPTICS AND INTERACTIVE SIMULATION FOR PRODUCT DESIGN REVIEW

Monica Bordegoni[(a)], Francesco Ferrise[(b)], Marco Ambrogio[(c)],
Giandomenico Caruso[(d)], Fabio Bruno[(e)], Francesco Caruso[(f)]

[(a) (b) (c) (d) (f)]Politecnico di Milano – Dipartimento di Meccanica – Milano (MI) - Italy
[(c) (e) (f)]Università della Calabria – Department of Mechanical Engineering – Rende (CS) - Italy

[(a)]monica.bordegoni@polimi.it, [(b)]francesco.ferrise@polimi.it, [(c)]marco.ambrogio@kaemart.it,
[(d)]giandomenico.caruso@polimi.it , [(d)]f.bruno@unical.it, [(d)]francesco.caruso@unical.it

ABSTRACT

The aesthetic impact of a product is an important parameter that makes the difference among products technologically similar and with same functionalities. The shape, which is strictly connected to the aesthetic impact, has different meanings if seen from the design and the engineering point of view. This paper describes an environment based on an integration of Mixed-Reality technologies, haptic tools and interactive simulation systems, named PUODARSI whose aim is to support designers and engineers during the phase of design review of aesthetic products. The environment allows the designer to modify the object shape, through the use of haptic devices, and the engineer to run the fluid-dynamics simulation on the product shape. The paper describes the main problems faced in integrating tools, originally developed for different purposes and in particular issues concerning data exchange, and the choice of those algorithms that guarantees low computational time as required by the application.

Keywords: Mixed-Reality, haptic interfaces, fluid-dynamics analysis, design review.

1. INTRODUCTION

It is well known that the design review process is time consuming, requires the collaboration and synchronization of activities performed by the two type of experts having different competences and roles, and is performed using different tools and different product representations. For this reason, new tools supporting the process (Szalavari, Schmalstieg, Fuhrmann and Gervautz 1998, Dani and Gadh 1997, Jayaram, Connacher and Lyons 1997)

Conceptual design of shape of aesthetic products is usually performed by designers. Design review is mainly performed according to requests for changes coming from engineering studies. Sometimes, modifications to product shape are done on digital representations, some other times they are performed on physical prototypes of the product.

Analyses of the product design, such as structural FEM (Finite Element Method) or CFD (Computational Fluid Dynamic) analysis, are performed by engineers. According to the results, it is very common that the shape of the object requires to be modified. The modification impacts the aesthetic value of the product, and therefore the designer is asked to take part to that.

Recent techniques and technologies of virtual prototyping allow users to answer to the requirements of the traditional design, as well as to industrial design; they are based on the definition of a digital model of the product including geometrical, topological and functional information. The aim of virtual prototyping is to provide design and validation tools, which are easy to use, which improve the product quality, and at the same time reduce the need of building physical prototypes, which are expensive, require time, do not allow easy changes, configurations and variants, and often do not allow validation and checking iterations. The virtual prototype is used from the initial phases of the design process to perform analysis and validation through simulation, in order to reduce the number of physical prototypes needed, and concentrate their use just at the end of the design process (Ulrich and Eppinger 2004)

This paper describes the PUODARSI environment whose aim is to support designers and engineers during the phase of design review of aesthetic products. The environment is based on Augmented Reality and haptic tools for improving the design review process. The basic idea is having a unique environment that can be used in a collaborative way by designers and engineers, where the object shape can be visualized in a realistic modality and at the same time by the two users. The environment allows the designer to modify the object shape easily and intuitively, through the use of haptic devices, and the engineer to perform the analysis and simulation on the new shape. These activities are reiterated until a consensus about the aesthetic and technical aspects of the product is reached. The evaluation of the various solutions can be performed in comparative way, and decisions can be taken simultaneously and collaboratively by the two experts.

The paper presents the conceptual description of the environment, the hardware and software technologies selected, and its implementation. The research has been carried out in the context of the PUODARSI (Product User-Oriented Development based on Augmented Reality and interactive Simulation) project, funded by the Italian Ministry of University and Research.

2. SYSTEM CONCEPTION

The PUODARSI environment is an interactive system that allows users to visualize in 3D or in an Augmented Reality (AR) environment the geometrical model of a product, to perform fluid-dynamics tests in real-time and to modify the geometrical shape of the object.

The system is intended to be used for design review and analysis of a product whose geometrical representation is available. The geometrical representation of the product can be reconstructed from a real existing object by means of reverse engineering techniques, or it can be created using a CAD tool. The object model is imported within the PUODARSI environment and shown to users in stereoscopy or through the use of AR technologies.

Two are the users of the system: designer and analyst. The designer can evaluate the aesthetic shape of a product; the analyst is in charge of performing fluid-dynamics analysis (CFD) of the product model. The result of the CFD analysis is shown through the visualization of the velocity field of air around the object. The model and the analysis data may be visualized in a stereoscopic modality; alternatively, analysis data may be visualized onto the physical prototype of the product by using an AR technology (Figure 1).

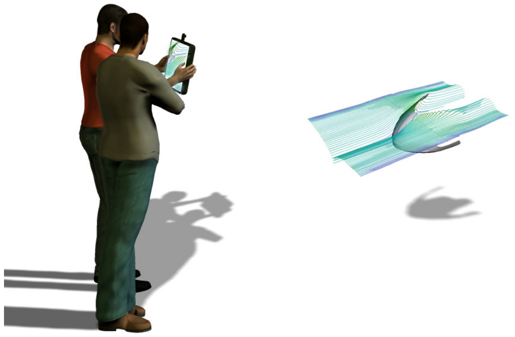


Figure 1: Conceptual image of the PUODARSI augmented reality environment where CFD analysis results can be seen super-imposed onto a physical prototype of the product.

The evaluation of fluid-dynamics characteristics of the model may reveal some problems (such as vortexes) that require a modification of the shape of the object in order to be fixed. Therefore, the analyst may decide to ask the designer to modify the shape providing specific information about the motivation and the area where to intervene. The designer can modify the shape of the product quickly, easily and intuitively using a haptic interface (Figure 2). The modification is performed on the basis of the requests made by the analyst, and also taking into account stylistic preferences.

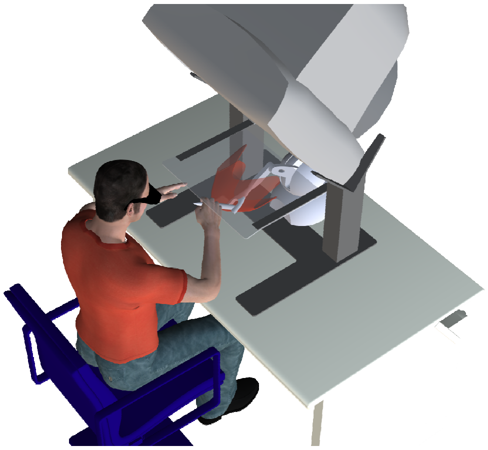


Figure 2: Conceptual image of the PUODARSI haptic design environment.

The PUODARSI environment can be though of as a simulation of a wind tunnel where one can simulate the interaction of air with an object in a virtual environment, and where the object shape can be modified directly and interactively as if one were inside the wind tunnel, but without actually perturbing the phenomena.

2.1. Theoretical background

The architecture of the environment is demanding especially for what concerns the performances required (for example, interactive haptic rendering, real-time simulation), and the integrability of hardware and software components.

For what concerns the haptic rendering, we have focused on the so called ’haptic modeling’ that addresses the modeling of virtual shapes using haptic technologies. The novelty of these technologies is that they allow users to touch, feel, manipulate and model objects in a 3D virtual environment that is similar to a real natural setting. Several applications based on haptic modeling have been developed (Burdea 1999).

Some applications have been developed with the aim of providing haptic interaction with volume dataset, without actually providing realistic force feedback (Avila and Sobierajski 1996, Iwata and Noma 1993). Some sculpting systems have been developed based on haptic force associated with dynamic subdivision of solids, which give users the illusion of manipulating semi-elastic virtual clay (Celniker and Welsh 1992, Dachille, Qin and Kaufman 2001, Jagnow and Dorsey 2002).

Our research group has developed a system based on haptic technology for the generation and evaluation of digital shapes for aesthetic purposes. The system has been developed in the context of the T’nD project - a project partially funded by the European Union and has been described in several papers (Bordegoni and Cugini 2007). A subsequent project is ongoing for developing a system based on a new concept of haptic interface for shape evaluation and supporting a stereoscopic ergonomic setup. This new system is being developed within the framework of the European project SATIN - Sound And Tangible Interfaces for Novel product design (Bordegoni, Covarrubias and Cugini 2008).

Regarding Computational Fluid Dynamic (CFD) applications, they allow users to investigate the behaviour of the fluids in a fixed volume by numerical solving the differential equations of Navier-Stokes, through a discretization process that uses the finite volume method (standard approach), the finite elements or the finite differences (Ferziger and Peric 1999, Anderson 1995). A CFD simulation is divided into three distinguished phases: pre-processing, simulation and post-processing. The first step consists of the definition of the control volume, the generation of the mesh, and the physical parameters of the fluid. All these data are passed to a solver that computes the solution.

Scientific visualization algorithms are used to visualize the results of the simulation that generally consist of datasets. The visualization application aims at presenting to users the simulation data in intuitive and direct way. Some research works have tried to create real-time simulations of the fluid behaviours in virtual dynamic fields (Chen, Lobo and Moshell 1997).

3. OVERVIEW OF HARDWARE AND SOFTWARE

The hardware and software components of the system have been decided upon a thorough analysis of the state of the art in the reference domains: 3D visualization, haptic interfaces and interactive simulation. Some choices concerning hardware components have been done considering the various technologies already available at our labs; software components have been selected among available open source libraries.

For what concerns visualization systems, several VR technologies have been considered (Dai 1998), such as:

- the full immersion stereoscopic HMD, 5DT 3D HMD 800; the augmented see-through NOMAD HMD; the Nvisor ST see-through HMD.
- projection-based systems, such as a system developed within the context of the European project SATIN (Bordegoni, Covarrubias and Cugini 2008), and the SenseGraphics system.
- the wall display system CyViz.

For what concerns open source software libraries supporting stereoscopic visualization, we have considered VTK, openSceneGraph and OpenSG. These libraries have been considered mainly because of the possibility of integrating external libraries and of importing dataset generated by simulation environments. Particularly interesting for our aims is VTK, which is a scientific visualization library that manages CFD structured and unstructured grids. VTK library has been successfully integrated with the Augmented Reality library ARToolKit. This AR library is named VTK4AR. This library has been used for the visualization of CFD data in an AR environment as described in (Bruno, Caruso, Ferrise and Muzzupappa 2006).

For what regards the haptic system for haptic modeling, we have considered the following haptic devices at our disposal in our labs: the Phantom device by SensAble, the HapticMaster by MOOG-FCS and the Virtuose system by Haption.

The following open source software libraries supporting real-time haptic rendering have been considered: CHAI3D, H3D, HaptikLibrary, OpenHaptic, and osgHaptics.

It has been decided to use the Phantom device mainly because all the libraries for haptic rendering support these devices since are very common and diffused. For what concerns the software library, it has been decided to use a generic haptic library, such as the H3D library, instead of using a proprietary library like the Sensable OpenHaptic, because it allows us to develop the application for the Phantom device, but also to run it with other haptic devices also supported by the library.

According to the requirements of the PUODARSI framework some issues have to be considered for what concerns the selection of the interactive simulation environment. One issue is related to the possibility of creating pre-processing steps that can be run automatically; a second issue concerns the compatibility of simulation results with an external scientific visualization environment.

At the conclusion of the analysis of the hardware and software it has been demonstrated that the STL file format is a good compromise for exchanging tessellated geometrical data among different software. The mesh generation algorithm of our system is expected to receive an STL geometric file and to generate a mesh that is compatible with the CFD solver. The solver should compute the solution in real-time, and also should give output results that are easy to manage by using the selected scientific visualization environment (VTK environment in our case). The software libraries that satisfy these requirements are the following: Deal II, OpenFlower, Comsol Multiphysics and OpenFOAM. Deal II is too limited since it does not support unstructured grids and works only with QUAD (2D) and HEX (3D) elements. OpenFlower is also written in C++, supports unstructured grids and requires GMSH as mesh generation algorithm. Comsol Multiphysics allows users to run multi-physics simulation. Some tests have been carried out on these libraries to verify the feasibility of performing real-time simulations (Ambrogio, Bruno, Caruso, Muzzupappa and Bordegoni 2008). Finally, OpenFOAM runs exclusively on Linux/UNIX operating systems. It receives in input a mesh in the GMSH format thanks to the gmshToFoam function, and generates an output file that is compatible with VTK thanks to the foamToVTK function.

4. PUODARSI ENVIRONMENT

The definition of the PUODARSI architecture has been defined upon the selection of hardware and software components made on the basis of the state of the art analysis.

The PUODARSI environment consists of the following main three modules:

1. module for haptic modification of product shape,
2. module for 3D rendering of object shape and of scientific data, and
3. module for interactive CFD analysis.

For what concerns the haptic interface used for shape modification, it has been decided to use the SenseGraphics 3D-IW immersive workbench that consists of a Phantom haptic device integrated with a stereo visualization system including a CRT monitor, a semi-transparent mirror and some stereographic shutter glasses. This configuration allows a good collocation of the haptic and the stereoscopic visualization space, and is more ergonomic compared to HMDs that users are not keen on wearing for long periods.

Regarding the software architecture, the major aspect that has been considered for the selection of the software libraries is the support for data exchange from the haptic application to the CFD simulation environment, and from CFD module to the data visualization in a Virtual/Augmented Reality environment. The selection of the libraries also considers that the initial file received in input is a CAD geometric file in STL format. For what concerns the selection of the haptic library, the environment should include a library that allows us to modify a tessellated CAD model. Actually, for the purpose of the project, that is mainly performing a feasibility study, a simple single contact point deformation algorithm based on a three-dimensional Gaussian curve is enough.

The following sections present the system architecture, its modules and their implementation and integration.

4.1. System architecture

On the basis of the hardware selection and of the system requirements and software benchmarks, the following software libraries have been selected:

- H3D haptic library for deforming tessellated models;
- OpenFOAM library for performing real-time CFD analysis;
- VTK visualization library for scientific data visualization.
- ARToolKit as pattern-based tracking system for the visualization module.

H3D library has been considered the most appropriate solution for haptic rendering since it already includes some useful algorithms for implementing tessellated surfaces deformation. OpenFOAM library allows us to manage input files from CAD modeling software through the GMSH module, and to export the results to VTK. ARToolKit is suitable for supporting the implementation of an augmented reality application integrated with the VTK library.

The system architecture is based on the use of different computers for the visualization/haptic interaction and the simulation system; this choice has been made mainly because the OpenFOAM library works only on Linux operating systems.

The Visualization System has been defined as the main component that exchanges data with all the other components and serves as rendering engine all the system components, as shown in figure 3. The integration of software components is mainly based on data exchange. All data pass through the VTK library that exchanges geometrical models in STL format with the haptic system and with the CFD solver. Moreover, it receives results generated from the CFD analysis (exported in VTK format), processes these data and visualizes them.

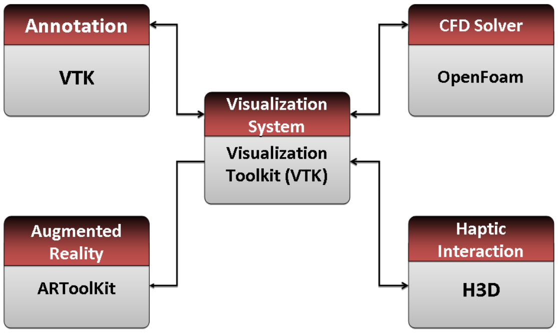


Figure 3: Software components of the PUODARSI environment.

4.2. System implementation

This section presents the PUODARSI environment, describing in details the implementation issues of its components and their connection.

4.2.1. CFD analysis module

The module for CFD analysis is implemented using the OpenFOAM library.

Since haptic libraries usually allow us to work easily especially with tessellated surfaces, it has been necessary to find a way to import tessellated geometries, create a control volume around these geometries to simulate the fluid volume and then generate a 3D mesh on the volume obtained by the subtraction of the volume of the model from the control volume.

For this purpose, the GMSH software library has been used for importing the geometry and generating the mesh. GMSH library works with .GEO files that are written in ASCII code using a very simple syntax; it can import STL files but does not work correctly when defining additional geometries (for example, the control volume). Therefore, a C++ routine for converting .STL files into .GEO file format has been implemented. The code works in such a way that, once the conversion is done, a control volume is created around the model.

After completing the definition of the geometry it is possible to generate the mesh (made of tetra elements) using the GMSH library. The output is a mesh file. Afterwards, the mesh can be converted in OpenFOAM format using the gmshToFOAM function included in the OpenFOAM package.

Once all information about the initial conditions is defined, the analysis can be launched using the solver routine icoFoam, a transient solver for incompressible, laminar flow of Newtonian fluids. Then the results are sent to the Visualization System that renders these data. The results of the analysis are transformed into a format that is compatible with VTK, using the function foamToVTK. This function creates a VTK file containing the definition of a VTK dataset and information about pressure and velocity for each vertex of the grid.

In our project VTK has been used to visualize geometry in STL format, and also the velocity field around the geometry that is obtained from the CFD analysis. By importing the dataset as a VTK unstructured grid it is possible to choose different modalities for representing a fluid flow field, such as stream tubes, stream ribbons, glyphs and so on, according to the preferences of the analyst that is performing the fluid-dynamics analysis. In the example reported in the following (Figures 4 and 5), we have decided to use streamlines to represent the velocity field. Streamlines are curves that are tangent to the velocity vector of the flow.

Figures 4 shows the results of the CFD analysis performed on an initial geometry of a windscreen; figure 5 shows the resulting analysis performed on the same object where the geometry has been modified using the haptic module.

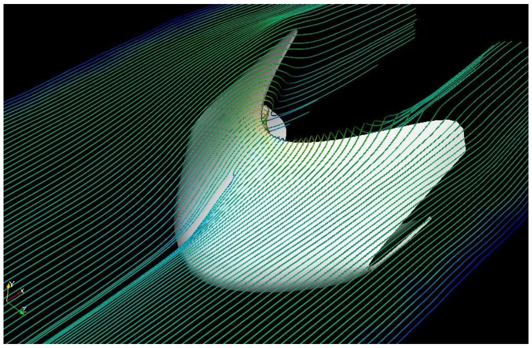


Figure 4: Fluid-dynamics analysis results performed on the initial geometry of a windscreen.

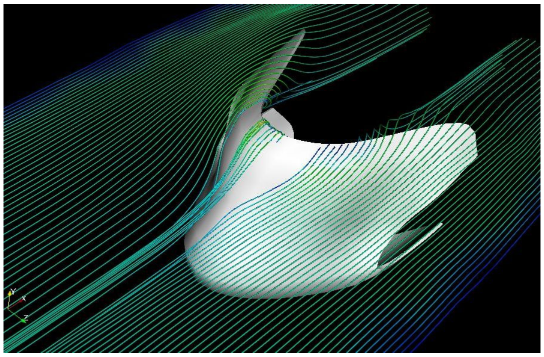


Figure 5: Fluid-dynamics analysis results performed on the modified geometry of a windscreen.

The VTK module is being integrated with the ARToolKit environment that supports Augmented Reality applications in a framework named VTK4AR (Bruno, Caruso, Ferrise and Muzzupappa 2006). This application allows users to see the object model and the analysis results in a stereoscopic environment.

4.2.2. Haptic shape modification module

The environment allows the designers to modify the object shape using a haptic module. The module is based on the SenseGraphics system consisting of an augmented reality visualization system, a haptic Phantom device and a 3D Connexion SpaceNavigator. The haptic shape modification module has been implemented using the H3D library.

H3D library works with geometries in the X3D format; since the model is initially available in STL format, it is necessary to create a converter from STL to X3D and from X3D to STL. The STL model is imported and transformed in X3D format, and then handles as a deformable shape (named DeformableShape). The haptic system deforms the geometry following a 3D Gaussian curve. The system provided users the possibility to change the width and the amplitude of the Gaussian function in order to define the area of influence of the deformation.

When the modification phase is finished, a new STL file representing the modified shape is generated; this model can be passed to the CFD module for running the analysis.

The model is visualized through the Visualization System. The system is based on the visual rendering library used by the haptic module that is VTK instead of H3D. The system includes a VTKrenderer function that has been specifically implemented for rendering 3D VTK objects (such as the output of the CFD module and annotations), X3D shapes and DeformableShapes. The visualization output is shown through the active stereoscopic window of the SenseGraphics system. Figure 6 shows a designer that is modifying the shape of a windscreen model using the haptic module.

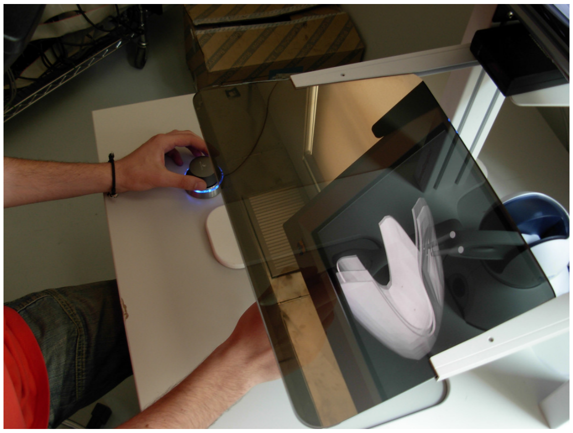


Figure 6: Use of haptic module based on the SenseGraphics system for the modification of the model of the windscreen.

4.2.3. Modules integration

Once the modules have been defined and implemented separately, it has been created a connection between them. The CFD application, using OpenFOAM library has been implemented on a machine running Linux operating system; the haptic module and the Visualization System, using H3D and VTK libraries, have been implemented on a machine running Windows operating system.

The haptic module and the Visualization System, and the CFD solver communicate through file exchange mode that is based on a Server/Client architecture using a TCP/IP connection (class ”socket”).

5. CONCLUSIONS

The paper has presented the preliminary results of the PUODARSI project aiming at developing an integrated environment for design reviews and real-time CFD analysis of aesthetic products. Some problems have been encountered during the selection of the components for the development of the environment, mainly from the software point of view, due to the exchange of information among the various components. Most of the components have been selected considering issue concerning integration among the various libraries rather than specific performances.

The system uses a STL file as an exchange shape data format, and the analysis and post-processing steps are completely independent from the modeling one. This will allow us in the future to develop the framework as an add-on for other modeling tools, such as the ones we are developing within the context of other two research projects, T’nD and SATIN projects, but also for the FreeForm system.

REFERENCES

Ambrogio M., Bruno F., Caruso F., Muzzupappa M., Bordegoni M., 2008. Interactive cfd simulation in virtual reality. *Proceedings of the XIX ADM Ingegraf* , June 4-6, Valencia.

Anderson J. D.,1995 *Computational Fluid Dynamics*, McGraw-Hill.

Avila R., Sobierajski L., 1996. A haptic interaction method for volume visualization, *IEEE CS*, Washington DC, 197-204.

Bordegoni M., Cugini U., 2007. Haptic interface for real-time evaluation and modification of shape design. *Proceedings of the ASME 2007 Design Engineering Technical Conferences & Computers and Information in Engineering Conference, IDETC/CIE*, September 4-7, Las Vegas, Nevada, USA.

Bordegoni M., Cugini U., Covarrubias M., 2008, Design of a visualization system integrated with haptic interfaces. *Proceedings of TMCE 2008 Conference,* Izmir, Turkey, April 21-25.

Bruno F., Caruso F., Ferrise F., Muzzupappa M., 2006. Vtk4ar: An object oriented framework for scientific visualization of cae data in augmented reality. *Proceedings of the Fourth Eurographics Italian Chapter Conference*, 75–81, Catania.

Burdea, G. C. 1999 Haptic Feedback for Virtual Reality, *Virtual Reality and Prototyping Workshop*, Laval (France).

Celniker G., Welch W 1992. Linear constraints for deformable b-spline surfaces. *Proceedings of the Symposium on Interactive 3D Graphics*, ACM N. Y., (Ed.), p. 165-170.

Chen, J. X., Lobo, N. D., Moshell, J. M. 1997. Real-time fluid simulation in a dynamic virtual environment. *IEEE Computer Graphics and Applications*, May-June.

Dachille F., Qin H., Kaufman A., 2001. A novel haptics-based interface and sculpting system for physics-based geometric design. *Computer Aided Design,* vol. 33, p. 403-420.

Dai F., 1998. *Virtual Reality for Industrial Applications*, Springer.

Dani, T. & Gadh, R., 1997 COVIRDS: *A Conceptual Virtual Design System* 29, 555 – 563.

Iwata H., Noma H. 1993. Volume haptization. *Proceedings of the IEEE Symposium on Research Frontiers in Virtual Reality* , pp. 16–23.

Jagnow R., Dorsey J., 2002. Virtual sculpting with haptic displacement maps. *Proceedings of Graphics Interface*.

Jayaram, S., Connacher, H. I. & Lyons, K.W., 1997. *Virtual assembly using virtual reality techniques*, 29, 575 – 584.

Szalavari, Z., Schmalstieg, D., Fuhrmann, A., Gervautz, M., 1998. Studierstube. An Environment for Collaboration in Augmented Reality Virtual Reality - *Systems, Development and Applications*, n. 1, 37 – 49.

Ulrich, K., Eppinger, S. D. 2004 *Product design and development*, third edition. Mc Graw Hill.

SIMULATION ANALYSES OF A NON-KINEMATICAL CONICAL ROLL BENDING PROCESS WITH CONICAL ROLLS

Zhengkun Feng [(a)], **Henri Champliaud** [(b)], **Thien-My Dao**[(c)]

Department of Mechanical Engineering, École de technologie supérieure, University of Quebec,
1100 Notre-Dame Street West, Montreal, Quebec, H3C 1K3, Canada

[(a)]feng.zhengkun@etsmtl.ca, [(b)]henri.champliaud@etsmtl.ca, [(c)] thien-my.dao@etsmtl.ca

ABSTRACT

In order to reuse the existing conical rolls of the kinematical conical roll bending process, attachments are added in the non-kinematical conical roll bending process to reduce the velocity of the top edge of the plate. In contrast with a kinematical conical roll bending machine, the driving outer rolls of a non-kinematical conical roll bending machine are allowed to slide on the zone close to the top edge of the plate due to the friction between the attachments and the plate. Therefore, the appropriate velocity obtained at the top edge of the plate makes it possible to roll the plate as using the kinematical conical roll bending process. This paper deals with the simulation analyses of the non-kinematical multi-pass conical roll bending process based on the finite element method. A well bent cone was obtained compared with the ideal cone and the numerical simulation results show that the bent cone depends on the static and dynamic friction coefficients for a geometrical configuration with an appropriate span of the outer rolls.

Keywords: conical roll bending, three-dimensional, Finite element method, Simulation

NOMENCLATURE

d	Span between the outer rolls
h	Real height of the desired cone
H	Whole height of the desired cone
l_i	Available length of the inner roll
l_o	Available length of the outer rolls
L_i	Whole length of the inner roll
L_o	Whole length of the outer rolls
r	Top radius of the desired cone
r_{pi}	Inside radius of the plate
r_i	Top radius of the inner roll
r_o	Top radius of the outer rolls
R	Bottom radius of the desired cone
R_{po}	Outside radius of the plate
R_i	Bottom radius of the inner roll
R_o	Bottom radius of the outer rolls
t	Plate thickness
ϕ	Angle of the desired cone
θ_{len}	Angular length of the plate
θ_{xy}	Angle with XY plane
θ_{xz}	Angle with XZ plane
θ_{yz}	Angle with YZ plane

1. INTRODUCTION

Large and long tubular sections used in industries are usually assembled with several segments and high geometrical quality of bent cones is required, such as the manufacturing of a crown of Francis turbines (Figure 1). For the purpose of satisfying the hydraulic profile (Fig. 1a), a crown is conventionally manufactured by foundry process which takes unexpected long time so the process becomes very expensive. An alternative process, which is capable of achieving an approximate hydraulic profile and accelerating the manufacture, was proposed (Marroquin 2002). With this process, a crown can be obtained by assembling several cone segments, for example, three cones of different size (Fig. 1b).

Both continuous conical roll bending and continuous cylindrical roll bending use usually three-roll or four-roll bending machines. Intensive research on the process of four-roll bending machines was undertaken for understanding the bending mechanism by Hua and his colleagues (Hua and Lin 1999a, Hua et al., 1994, Hua, Baines, and Cole 1995, Hua, Sansome, and Baines 1995, Hua et al., 1997, Hua, Baines, and Cole 1999, Hua and Lin 1999b, Lin and Hua 2000). Such bending machines have the advantage in eliminating the planar zones close to the leading and trailing edges of the plate. This advantage is significant if the plate is thin in cylindrical roll bending process. However, it is not efficient for conical bending (Hua and Lin 1999a, Roggendorff and Haeusler 1979), especially if the plate to be bent is thick. In addition, the process is more complex than that of the three-roll bending machines, the pre-bending for eliminating the planar zones needs to position the plate repeatedly. Although analyses of pyramidal three-roll bending process in cylindrical roll bending as two-dimensional modeling can be found in literature (Roggendorff and Haeusler 1979, Yang, Mori, Osakada 1994, Yang and Shima 1988, Ramamurti, Rao, Sriram 1972, Bassett and Johnson 1966, Gandhi and Raval 2008), it is necessary to apply more complete theory to a three-dimensional model of conical roll bending process.

With the fast evolution of computer science and technology, powerful computational technology makes such applications possible. For example, the finite element method is widely used in solid mechanics and commercial software is easily accessible for efficient analyses through numerical simulations, such as the well-known commercial software ANSYS/LS-DYNA. Three-dimensional modeling based on finite element method can be found recently in literature, such as the modeling and simulation of kinematical conical roll bending process (Zeng, Liu, and Champliaud 2008) and non-kinematical conical roll bending process with conical rolls (Feng et al. 2007, Feng, Champliaud, and Dao 2008).

This paper deals with the simulation analyses of the non-kinematical multi-pass conical roll bending process with conical rolls based on the finite element method. The remaining sections of this paper are organized as follows: Section 2 presents the operational sequences of the symmetrical pyramidal three-roll bending process; Section 3 describes the geometrical configurations of the three-dimensional model; Section 4 discusses the model based on finite element method. Section 5 presents the numerical simulation results with discussion, and finally, conclusions will be drawn.

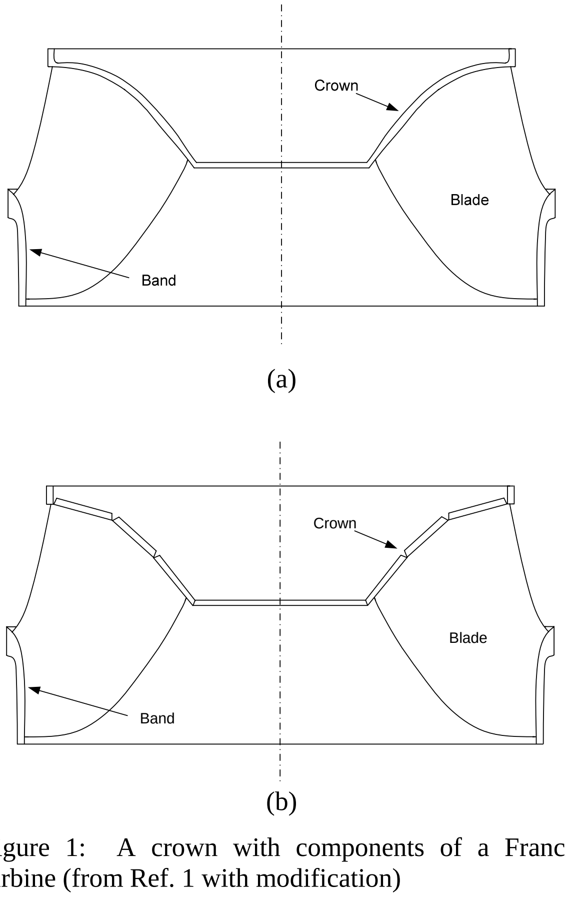


Figure 1: A crown with components of a Francis turbine (from Ref. 1 with modification)

2. OPERATIONAL SEQUENCES

Figure 2 illustrates the front view of the geometrical configuration of a simplified model of a symmetrical pyramidal three-roll bending machine. The model consists essentially of one plate, one inner roll and two outer rolls. The two identical outer rolls with a constant distance between their axes and in a symmetrical position related to the inner roll are driving rolls while the inner roll is able to move in the vertical direction in order to press down the plate to reach a desired curvature. The dot lines represent the initial positions of the plate and the inner roll while the continuous lines represent the deformed plate and the final position of the inner roll. The four small vertical cylinders are used as guides to keep the plate from moving away from the bending machine. The two small horizontal semi-cylinders are used as attachments (see zoom without the guiding cylinders). The distances between the plate/attachment contact lines (the contact lines between the plate and the attachments) and the outer contact lines (the contact lines between the plate and the outer rolls) are exactly the thickness of the plate. During the roll bending process, the inner roll presses the plate down to a desired position at point B, where the inner contact line (the contact line between the inner roll and the plate) passes and in the meanwhile, the plate is driven by the rotation of the two outer rolls. The curvature of the deformed plate at point B is supposed to be the desired curvature of the desired cone. It decreases progressively from point B to point A and point C, respectively. The zones of the plate between the leading edge and the outer contact line (point A) at the beginning and between the trailing edge and the inner contact line (point C) at the end of the process will never pass under the inner roll, and will stay planar (In fact, it is not strictly planar due to the deformation produced by the pressing of the inner roll on the inner contact line).

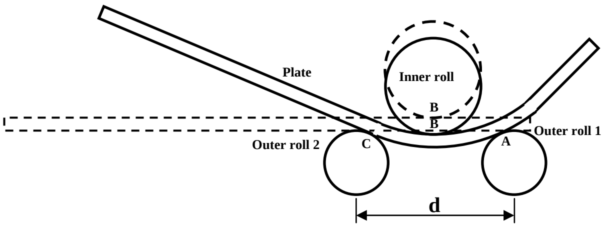


Figure 2: Front view of a symmetrical pyramidal three-roll bending machine

3. GEOMETRICAL CONFIGURATION

Figure 3 illustrates the initial geometrical configurations of the non-kinematical conical bending model. The process is determined by the final position of the desired cone with some parameters provided by customers and engineers, such as the top and bottom radii, the height of the desired cone and the thickness of the plate by customers and the top and bottom radii of the rolls and the span of the outer rolls (the distance between the axes of the outer rolls) by engineers. With these parameters, the other necessary geometrical parameters of the model are calculated in the following section. Since the axes of the two outer rolls can only rotate around their axes during the roll bending process, their initial positions, which are also their final positions, will be determined first, then the initial position of the plate, and finally, those of the inner roll, will be determined.

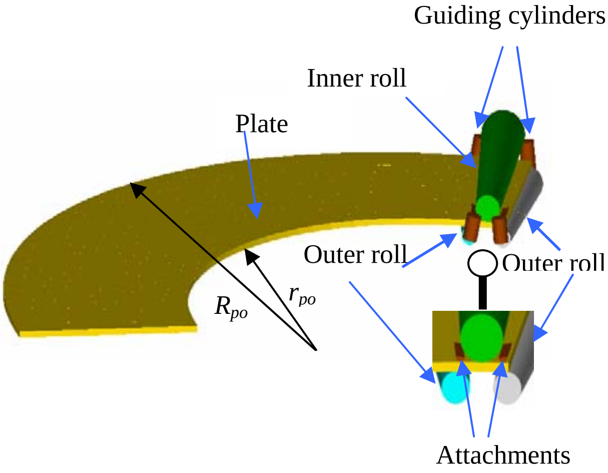


Figure 3: Initial positions of the components of a pyramidal three-roll bending machine in the non-kinematical roll bending process with conical rolls

3.1. Position of the outer rolls

The bottom center of the cone is chosen as the origin of the global Cartesian system, where axis Z coincides with the axis of the desired cone (Fig. 4). The geometry of the final cone determines the position of the outer rolls, where the cone angle ϕ is determined by its height, its top and bottom radii, as follows:

$$\phi = \tan^{-1}\left(\frac{R-r}{H}\right) \tag{1}$$

The position of the apex of the outline of the cone locates at P(0, 0, R/tanϕ) and the angle between the axis of an outer roll and the outer contact lines is:

$$\beta = \sin^{-1}\left(\frac{(R_o - r_o)\sin\phi}{R-r}\right) \tag{2}$$

Adding the two previous angles yields the angle between the cone axis and the axis of each outer roll as follows:

$$\alpha = \phi + \beta \tag{3}$$

The available length of the outer rolls, which are truncated cones, is:

$$l_o = L_o\left(1 - \frac{r_o}{R_o}\right) \tag{4}$$

where L_o is the whole length of the outer rolls (the shape of non-truncated cones)

$$L_o = \frac{R_o}{\tan\beta} \tag{5}$$

The angles between the outer rolls and the XY, YZ, XZ planes can be obtained from the following equations:

$$\theta_{xyo} = \sin^{-1}\left(\frac{d\sin\beta}{2(L_o\sin\alpha\sin\beta + R\sin\beta - R_o\sin\phi)}\right) \tag{6}$$

$$\theta_{yzo} = \tan^{-1}\left(\tan\alpha\cos\theta_{xyo}\right) \tag{7}$$

$$\theta_{xzo} = \tan^{-1}\left(\tan\alpha\sin\theta_{xyo}\right) \tag{8}$$

The apexes of the outer rolls locate at points $P_{o1}(x_{o1}, y_{o1}, z_{o1})$ and $P_{o2}(x_{o2}, y_{o2}, z_{o2})$, respectively, where

$$x_{o1} = -x_{o2} = \left(\frac{R_o}{\sin\beta} - \frac{R}{\sin\phi}\right)\sin\phi\sin\theta_{xyo} \tag{9}$$

$$y_{o1} = y_{o2} = \left(\frac{R_o}{\sin\beta} - \frac{R}{\sin\phi}\right)\sin\phi\cos\theta_{xyo} \tag{10}$$

$$z_{o1} = z_{o2} = H + \left(\frac{R_o}{\sin\beta} - \frac{R}{\sin\phi}\right)\cos\phi \tag{11}$$

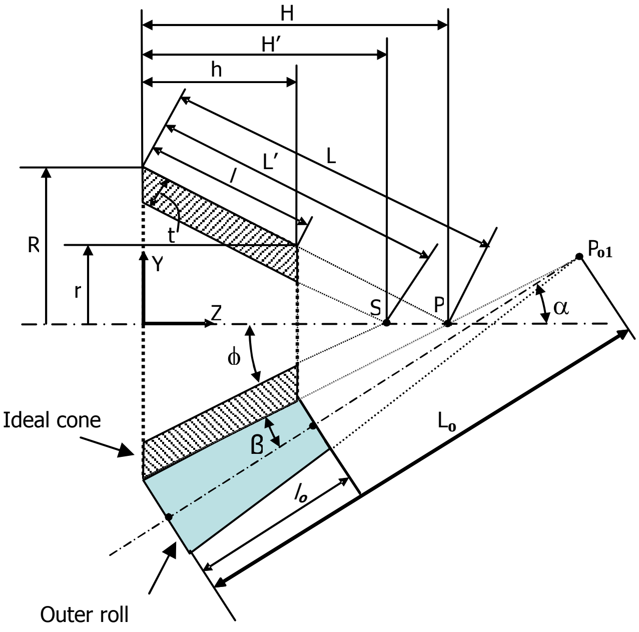


Figure 4: Positions of the desired cone and the conical outer rolls

3.2. Initial position of the plate

As the plate to be bent is determined by the desired cone or the ideal cone, the outside and inside radii of the plate (Fig. 3) are determined by the cone angle, the top and bottom radii of the cone and its height as follows:

$$R_{po} = \frac{R}{\sin\phi} \tag{12}$$

$$r_{po} = \frac{H-h}{\cos\phi} \tag{13}$$

The angles of the plane of the initial plate with the XY, XZ and YZ planes are:

$$\theta_{xyp} = \theta_{xzp} = 0 \tag{14}$$

$$\theta_{yzp} = \tan^{-1}\left(\tan(\alpha-\beta)\cos\theta_{xyo}\right) \tag{15}$$

The center of the plate locates at point (0, 0.5tcosθ_{xyo}, H-0.5tsinθ_{xyo}) and the angular length of the plate is:

$$\theta_{len} = \frac{2\pi R}{R_{po}} \tag{16}$$

3.3. Position of the inner roll

The initial and final positions of the inner roll, the initial position of the plate and the final position of the bent cone are illustrated in Figure 5. The position of the inner apex of the cone locates at point S(0, 0, H') (Fig. 4), where H'=H-t/sinϕ. The angle of the inner roll with the outline of the desired cone is:

$$\beta' = \sin^{-1}\left(\frac{(R_i - r_i)\sin\beta}{(R_o - r_o)}\right) \tag{17}$$

So the angle between the inner roll axis and the cone axis becomes:

$$\alpha' = \phi - \beta' \tag{18}$$

The available length of the inner roll, which is also a truncated cone, is:

$$l_i = L_i\left(1 - \frac{r_i}{R_i}\right) \tag{19}$$

where L_i is the whole length of the inner roll (the shape of a non-truncated cone)

$$L_i = \frac{R_i}{\tan\beta'} \tag{20}$$

The initial angles of the inner roll with the XY, XZ and YZ planes can be obtained from the following equations:

$$\theta_{xyi0} = \theta_{xzi0} = 0 \tag{21}$$

$$\theta_{yzi0} = \theta_{yzp} - \beta' \tag{22}$$

The initial position of the apex of the inner roll locates at point $P_{i0}(0, y_{i0}, z_{i0})$, where

$$\begin{aligned} y_{i0} = {} & (L_i\cos(\theta_{yzp}-\beta') - R_i\sin(\theta_{yzp}-\beta') \\ & - t\sin\theta_{yzp} - H')\tan\theta_{yzp} \end{aligned} \tag{23}$$

$$\begin{aligned} z_{i0} = {} & L_i\cos(\theta_{yzp}-\beta') \\ & - R_i\cos(\theta_{yzp}-\beta') - t\sin\theta_{yzp} \end{aligned} \tag{24}$$

Similarly, the final angles of the inner roll with the XY, XZ and YZ planes can be obtained from the following equations:

$$\theta_{xyi} = \theta_{xzi} = 0 \tag{25}$$

$$\theta_{yzi} = \phi - \beta' \tag{26}$$

The final position of the apex of the inner roll locates at point $P_i(0, y_i, z_i)$, where

$$\begin{aligned} y_i = {} & (L_i\cos(\phi-\beta') \\ & - R_i\sin(\phi-\beta') - t\sin\phi - H')\tan\phi \end{aligned} \tag{27}$$

$$z_i = L_i\cos(\phi-\beta') - R_i\cos(\phi-\beta') - t\sin\phi \tag{28}$$

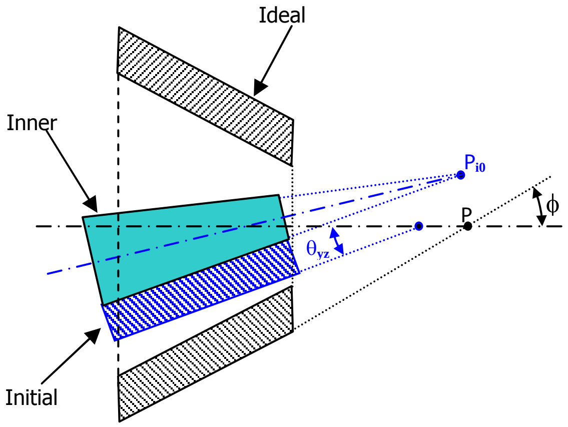


Figure 5: Initial positions of the inner roll and the plate and the final position of the plate (the bent cone)

4. FINITE ELEMENT MODELLING

Although finding a model capable of accurately to represent the roll bending process is the objective of this research, the following assumptions were applied for the purposes of computational efficiency:

- For the plate, the material is considered isotropic, with bilinear elastoplastic deformation behaviour (see the stress–strain curve in Figure 6), with constant properties for the elastic modulus E, the yield stress σ_Y and the tangent modulus E_T.
- The static and dynamic friction coefficients are constant on the plate/attachment contact lines and on the outer contact lines, zero on the inner contact line.
- The rolls, the attachment and the guiding cylinders are rigid bodies.
- The weight of the plate is not considered.

During the roll bending process, a downward force is applied on the inner contact line through the inner roll, and the neighbor zone of the plate encounters an elastic deformation (section OA) (Fig. 6) at first, followed by plastic deformation (section AB). This zone shifts from the leading edge to the trailing edge of the plate as the two outer rolls drive the plate to move forwards and the deformed zone attempts to return to its original state due to the springback. However, the plate stays finally at its equilibrium position at point D. The modeling and the numerical simulations are performed with ANSYS/LS-DYNA based on finite element method with explicit scheme. The finite element model uses the four-node shell elements of type shell163 to descritize the space domain. The governing equations of a deformable body in motion can be described as follows:

$$\sigma_{ij,j} + \rho f_i = \rho \ddot{x}_i \tag{29}$$

with the following boundary conditions:

$$\sigma_{ij} n_i = t_i(t) \tag{30}$$

The discretization of equations (29) and equations (30) based on the finite element method yields the following matrix equations:

$$\mathbf{M}\ddot{\mathbf{x}}(t) + \mathbf{F}(\mathbf{x},\dot{\mathbf{x}}) - \mathbf{P}(\mathbf{x},t) = 0 \tag{31}$$

The following explicit time integrations are applied to these semi-discretized equations to find sequentially the solutions (Hallquist 2005).

When the central difference time integration is applied to equations (31), the acceleration at time n can be obtained as

$$\ddot{\mathbf{x}}^n = \mathbf{M}^{-1}[\mathbf{P}^n - \mathbf{F}^n] \tag{32}$$

The velocities at time n+1/2 and the coordinates at time n+1 are explicitly updated as

$$\dot{\mathbf{x}}^{n+1/2} = \dot{\mathbf{x}}^{n-1/2} + \Delta t^n \ddot{\mathbf{x}}^n \tag{33}$$

$$\mathbf{x}^{n+1} = \mathbf{x}^n + \Delta t^{n+1/2} \dot{\mathbf{x}}^{n+1/2} \tag{34}$$

This approach has advantage for modeling structural dynamic system with large displacements and small deformations.

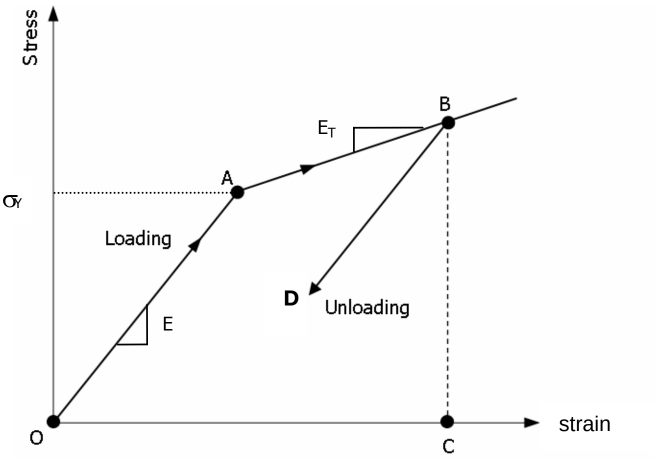


Figure 6: Simplified relation between the stress and strain of the plate

5. NUMERICAL SIMULATIONS

The numerical simulations were performed with a model based on finite element method described in the previous section. The material of the plate was assumed to be ASTM A36 steel. The shell elements (shell163 in ANSYS/LS-DYNA) were used to discretize the plate by 10 elements in the radial direction and 40 elements in the circumferential direction (Fig. 7). The other components of the roll bending machine were modeled with rigid four-node shell elements in order to save computational time. The geometrical parameters of the model and the properties of the ASTM A36 steel at the forming temperature of 400 ^{o}C are listed in Table 1. The radius ratio between the top and bottom of the rolls was 0.65 which is greater than 0.6 of the kinematical conical roll bending process, while the static and dynamic friction coefficients on the outer contact lines were 0.6 and 0.3, respectively. The process was a multi-pass with three passes. For a configuration as determined in section 2 with the parameters provided in Table 1, the final shape of the cone depended on the span of the outer rolls d, the static and dynamic friction coefficients F_{sa} and F_{da} on the plate/attachment contact lines. For this case, a well closed cone with a small gap between the leading and trailing edges was obtained when d was 0.992 m, F_{sa} and F_{da} were 0.8 and 0.4, respectively (Fig. 8). To verify the quality of the bent cone, firstly, the helical distortion of the top and bottom edges and the gap between the leading and trailing edges of the bent cone is analyzed with the parameters illustrated in Figure 9, where d1 is the distance between points B_1

(the intersection point of the bottom edge with the trailing edge of the final cone) and B_2 (the intersection point of the bottom edge with the leading edge of the final cone), similarly, d2 is the distance between points S_1 (the intersection point of the top edge with the trailing edge of the final cone) and S_2 (the intersection point of the top edge with the leading edge of the final cone). d3 and d4 are used for the measurement of distortion in the circumferential direction at the top and the bottom of the cone, respectively. d5 and d6 are used for the measurement of the helical distortion in the corresponding perpendicular direction (the direction parallel to the leading edge). The deviations of the helical distortion and the gap of the final cone are shown in Table 2. The helical distortion is as very small as 1 cm and the gap is also small. Secondly, the verification was performed with the comparison of the bent cone to an ideal cone. A fitted cone was obtained from the bent cone (Feng et al. 2007) (Fig. 10) and compared with the ideal cone as given in Table 1 (Fig. 11). The deviations of the top and bottom radii, the height and the angle of the fitted cone from the ideal cone were calculated as follows:

$$r_{err} = abs\left(\frac{r_f - r}{r}\right) \tag{35}$$

$$R_{err} = abs\left(\frac{R_f - R}{R}\right) \tag{36}$$

$$h_{err} = abs\left(\frac{h_f - h}{h}\right) \tag{37}$$

$$\phi_{err} = abs\left(\frac{\phi_f - \phi}{\phi}\right) \tag{38}$$

where r_{err}, R_{err}, h_{err} and ϕ_{err} are the deviations of the top and bottom radii, the height and the angle of the fitted cone from the ideal cone, r_f, R_f, h_f and ϕ_f are the top and bottom radii, the height and the angle of the fitted cone. These deviations are shown in Table 3 with the largest value of less than 8% occurring on the top radius of the fitted cone.

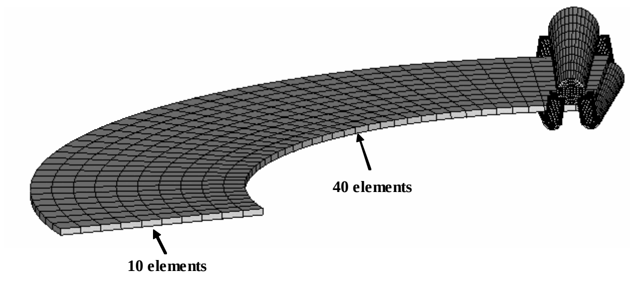


Figure 7: Mesh of the model with shell elements

Table 1: Geometrical parameters of the bending model and the material properties of the plate

Geometrical parameters:	
Cone:	
Height (m)	2.5
Top radius (m)	1.5
Bottom radius (m)	3.0
Inner roll:	
Bottom radius (m)	0.375
Radius ratio	0.65
Outer roll:	
Bottom radius (m)	0.25
Radius ratio	0.65
Material properties of steel ASTM A36 (400°C):	
Elastic modulus (GPa)	166
Yield stress (MPa)	562
Tangent modulus (GPa)	1.6
Poisson's ratio	0.3
Density (kg/m^3)	7850

Table 2: Deviations of the helical distortion and the gap of the final cone

d3(cm)	d4 (cm)	d5 (cm)	d6 (cm)
12.4	12.5	1.1	0.9

Table 3: Deviations of the fitted cone from the ideal cone

r_{err} (%)	R_{err} (%)	h_{err} (%)	ϕ_{err} (%)
7.8	4.56	0.0556	0.0058

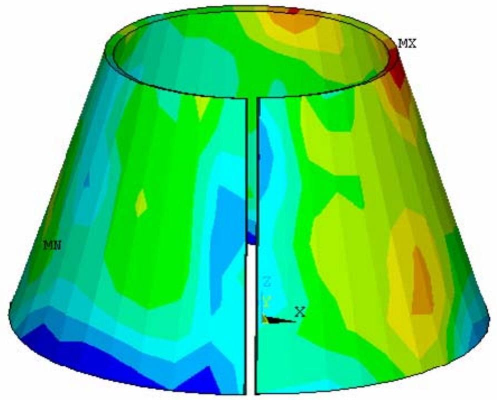


Figure 8: A bent cone obtained from non-kinematical conical roll bending with conical rolls

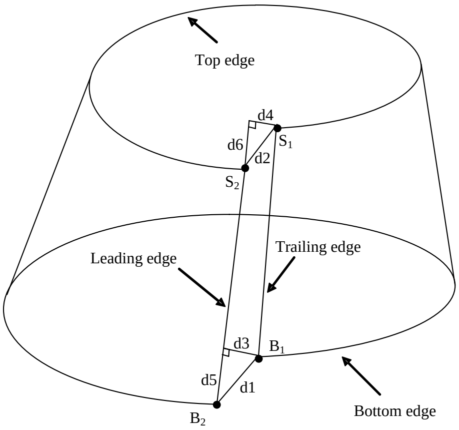


Figure 9: Parameters for determining the helical distortion of the final cone

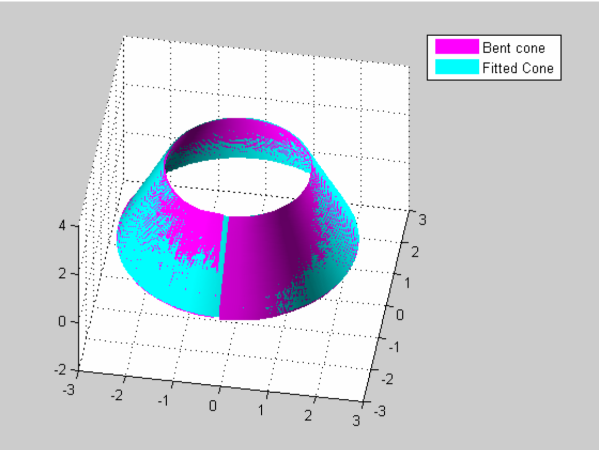


Figure 10: Fitted cone obtained from the bent cone

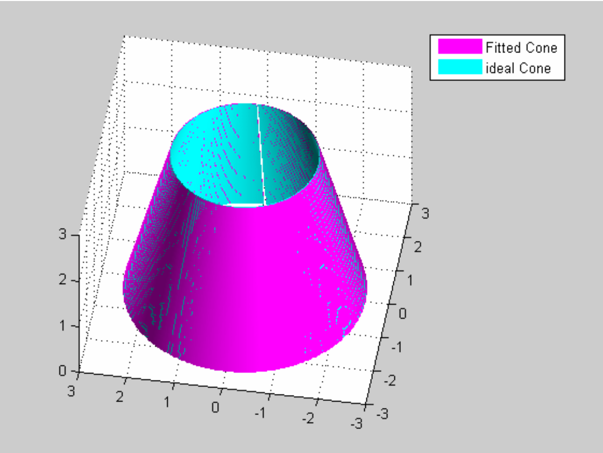


Figure 11: Fitted cone compared with the ideal cone

6. CONCLUSIONS

Continuous non-kinematical conical roll bending process with conical rolls has the advantage of reusing the existing conical rolls of kinematical conical roll bending process. For a determined geometrical configuration, the velocity ratio between the top and bottom edges of the plate is determined by the ratio of the top and bottom radii of the rolls together with the friction at the plate attachment contact. Therefore, the bent cone depends on the static and dynamic friction coefficients on the plate/attachment contact lines and the span between the two outer rolls. When using the existing inner and outer rolls with a determined ratio between the top and bottom radii, the desired closed cone can be obtained by adjusting the static and dynamic friction coefficients at the plate/attachment contact lines and the span between the two outer rolls. This process is less expensive and more adaptive than the kinematical roll bending process due to the possibility of the reuse of the existing rolls. However, these roll bending machines are not capable of eliminating the planar portions close to the leading and trailing edges of the plate as the kinematical roll bending process of pyramidal three-roll bending machines. Eliminating these planar zones by repeating the roll bending process after welding the closed cone remains one of the goals to be pursued in the future.

ACKNOWLEDGMENTS

The authors express their thanks to the Natural Sciences and Engineering Research Council (NSERC) of Canada and General Electric Canada for their financial support for this research.

REFERENCES

Bassett, M.B., Johnson, W., 1966. The bending of plate using a three roll pyramid type plate bending machine. *Journal of Strain Analysis for Engineering Design*, I(5), 398-414.

Feng, Z., Champliaud, H., Dao, T.M., 2008. Modeling and Simulation of Non-Kinematic Conical Roll Bending Process with Conical Rolls. Submitted to *Journal of Manufacturing Science and Technology*.

Feng, Z., Champliaud, H., Zeng, J., Dao, T.M., 2007. Investigation of Influence of Sliding of Rolls over Plate on Continuous Conical Bending. *Proceedings of the 18th IASTED International Conference on Modelling and Simulation, MS 2007*, pp. 527 – 531. May 30 - June 1, Montreal (Quebec, Canada).

Gandhi, A.H., Raval, H.K., 2008. Analytical and empirical modeling of top roller position for three-roller cylindrical bending of plates and its experimental verification. *Journal of Materials Processing Technology*, 197(1-3), 268-278.

Hallquist, J.O., 2005. *LS-DYNA theory manual*, Livermore Software Technology Corporation.

Hua, M., Baines, K., Cole, I.M., 1995. Bending mechanisms, experimental techniques and preliminary tests for the continuous four-roll plate bending process. *Journal of Materials Processing Technolog*, 48(1-4), 159-172.

Hua, M., Baines, K., Cole, I.M., 1999. Continuous four-roll plate bending: a production process for the manufacture of single seamed tubes of large and medium diameters. *International Journal of Machine Tools and Manufacture*, 39(6), 905-935.

Hua, M., Cole, I.M., Balnes, K., Rao, K.P., 1997. A formulation for determining the single-pass mechanics of the continuous four-roll thin plate bending process. *Journal of Materials Processing Technology*, 67(1-3), 189-194.

Hua, M., Lin, Y.H., 1999a. Effect of strain hardening on the continuous four-roll plate edge bending process. *Journal of Materials Processing Technology*, 89-90, 12-18.

Hua, M., Lin, Y.H., 1999b. Large deflection analysis of elastoplastic plate in steady continuous four-roll bending process. *International Journal of Mechanical Sciences*, 41(12), 1461-1483.

Hua, M., Sansome, D.H., Baines, K., 1995. Mathematical modeling of the internal bending moment at the top roll contact in multi-pass four-roll thin-plate bending. *Journal of Materials Processing Technology*, 52(2-4), 425-459.

Hua, M., Sansome, D.H., Rao, K.P., Balnes, K., 1994. Continuous four-roll plate bending process: Its bending mechanism and influential parameters. *Journal of Materials Processing Technology*, 45(1-4), 181-186. In French.

Lin, Y.H., Hua, M., 2000. Influence of strain hardening on continuous plate roll-bending process. *International Journal of Non-Linear Mechanics*, 35(5), 883-896.

Marroquin, L.R.H., 2002. *Méthodologie de design et de fabrication de la couronne d'une roue Francis*. Mémoire de Maîtrise, Département de Génie Mécanique, Université du Québec à Montréal.

Ramamurti, V., Rao, V.R.S., Sriram, N.S., 1992. Design aspects and parametric study of 3-roll heavy-duty plate-bending machines. *Journal of Materials Processing Technology*, 32(3), 585-598.

Roggendorff, S., Haeusler, J., 1979. Plate bending: three rolls and four rolls compared. *Welding and Metal Fabrication*, v. 47(6), 353-357.

Yang, G., Mori, K.I. Osakada, K., 1994. Determination of forming path in three-roll bending using FEM simulation and fuzzy reasoning. *Journal of Materials Processing Technology*, 45(1-4), 161-166. In Japanese.

Yang M., Shima, S., 1988. Simulation of pyramid type three-roll bending process. *International Journal of Mechanical Sciences*, 30(12), 877-886.

Zeng, J., Liu, Z., Champliaud, H., 2008. FEM dynamic simulation and analysis of the roll-bending process for forming a conical tube. *Journal of Materials Processing Technology*, 198(1-3), 330-343.

Library for Dynamic Simulation of Reverse Osmosis Plants

Luis G. Palacin*, Cesar de Prada, S. Syafiie**, Fernando Tadeo****
**Centro de Tecnologia Azucarera, Edificio Alfonso VIII, 47002 Valladolid, Spain*
*** Dpto. Ingenieria de Sistemas y Automatica, Fac. Ciencias, Univ. Valladolid, 47002 Valladolid, Spain*
(e-mail: fernando@autom.uva.es)

Abstract: This paper presents a library of simulated components for dynamic simulation of Reverse Osmosis plants. This library has been developed with the objective of testing different control configurations, and fault-detection modules. It consists of a set of simulated components that can be graphically connected to simulate efficiently whole plants. This makes possible to predict quite precisely the evolution of variable of interest for controller design.

---

1. INTRODUCTION

To combat water scarcity desalination activities are been intensive introduced in arid regions. One of the most popular strategies is the installation of RO desalination plants, which provide a cost-effective and simple solution for desalination (Wilf, 2007).

However, RO are by nature energy intensive, so good control is basic to maintain water-production costs at acceptable level and to elevate the plant availability, particularly in regions with high water scarcity or when removable energy sources are used. Thus, it is crucial to have available dynamic simulation tools so that advanced techniques of control can be tested (Alatiki, 1999; Gambier *et al.*, 2007). It has already been mentioned that the lack of dynamic models and simulation tools limits the application of advanced control in RO plants (Robertson *et al.*, 1996; Gambier and Badreddin, 2004).

Until now, cost reduction and energy saving have been undertaken almost exclusively from the point of view of technological improvement of the basic components of the plant: membranes and pumps (See, for example, Geislera, 2001; Seibert et al., 2004; Stover, 2004). As in many other engineering areas, technological advances in the plant components should be accompanied by a sophisticated control system design, for what adequate simulation tools, such as the one presented in this paper, are needed.

2. REVERSE OSMOSIS PLANTS

The RO process consists in extracting the water from a saline solution by pumping it against a membrane so that it is separated from the solutes (the dissolved material). For this, a differential pressure must be applied, higher than the osmotic pressure of the solution. The water that passes through the membrane (permate) is nearly pure, while the remaining water, with a high salt concentration (brine) is carefully discharged (See Wilf, 2007 for details on the process).

A typical reverse osmosis plant is shown in Figure 3: it can be seen that, apart from the membranes there are many other additional components needed to keep the system running and the water potable. These systems must be included in the simulation in detail, as they affect significantly performance and are the source of many faults in the real plant. From those we emphasize:

- *Pretreatment*, consisting of filtration and addition of chemical products to eliminate microorganisms and prevent precipitation in the membranes.
- *Postreatment*, to potabilize the water produced. This is usually carried out by adding chloride and increasing slightly the salinity, by mixing with filtered salty water.

In the last years, significant advances in membrane technology have allowed an improvement in the filtering quality and simultaneous reduction of costs. Hence, today RO plants demand less energy, investment cost, space requirements and maintenance than other desalination processes (Gambier *et al.*, 2007), so they are been extensively implanted in fresh-water depleted areas, specially combined with clean sources of energy (solar, wind, etc).

3. LIBRARY FOR SIMULATION OF RO PLANTS

3.1. *Objectives*

The objective was to develop a dynamic simulation of the RO that can be used to:

- decide about design specifications of the different components of the systems, comparing the different alternatives, not only in steady-state but also in transient conditions

- design and tune the control system based on the simulation under different environmental conditions
- design and test the fault detection module, simulating the presence of different faults and malfunctions.

Thus, the first objective was to develop computer models of the different components of the plant that can be used for dynamic simulation of RO plants to test different configurations, and use them to test fault detection and accommodation algorithms. These models have been presented in detail elsewhere (Syafiie, 2008), so this paper presents only the library developed within the EcoSim library.

3.2. *Library components*

The following components typical in a RO plant are included in the library (see Fig. 1 for icons used in the library):

- RO membranes
- Filters (Sand, Cartridges,…):
- Energy Recovery (Isobaric Chambers, Pelton Turbines, etc)
- Pumps (Centrifugal, Positive-displacement, etc)
- Storage tanks, pipes, valves, etc.

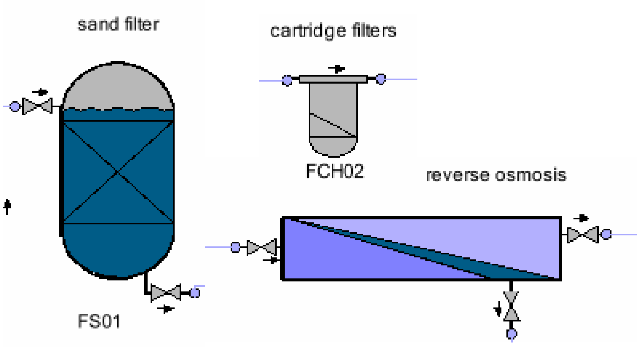


Fig. 1: Graphical representation of different components of the RO Library

3.3. *Input/output variables*

To make possible any interesting interconnection, components in the library are designed to be connected through ports with the following variables:

1) Flow
2) Pressure
3) Salt concentration
4) Concentration of solids, to reproduce faults in filters and membranes.
5) Temperature, to reproduce the effect of water temperature in membrane efficiency (Gambier *et al.*, 2007) and cleaning (Clayton, 1999).
6) An important aspect of RO plant is the power consumption, so pumps and energy recovery systems provide as additional output the power needed to operate, so the library can be used to check different strategies to reduce operation costs.
7) Additional variables are provided for the user to reproduce variables used for process monitoring, such as the Normalized Permeate Flow, Normalized Pressure Drop and Normalized Pressure Passage, used to schedule cleanings and detect critical faults.

3.4. *Methodology*

Modeling was based on using first-principles and empirical models from the literature. These models are then expressed directly using EcoSimPro, as this Simulation Software offers an efficient object-oriented solution to industrial modelling problems, generating directly C++ code that can be used to simulate the RO plant, or interfaced with standard control software.

4. DYNAMIC SIMULATION

There are several simulation tools commercially available, usually provided by membrane manufacturers. However these tools represent only static simulations, so they are very useful for design (sizing, configuration, layout, etc), but they can not be used for testing transient behaviour when

- switching between operational modes (standby/normal operation/membrane cleaning/filter cleaning/shutdown, etc) .
- planning cleaning times
- designing controllers
- designing fault detection algorithms

Compared with this, the developed library can be used for precise dynamic simulation, including starting-up, shutdown and the effect of cleaning and aging. For this, components can be directly connected to simulate whole plants (see an example Figure 3). More important, the dynamic simulations can be easily used to check the responses of the system in different situations: For example, Figure 4 shows the evolution of the permeability of the cartridge filters when a high concentration of fine solids is present in the inlet (typical of surface water): if it were not filtered the filter would clog.

Another example of the possibilities of the library developed is shown in Figure 5, which depicts the evolution of the different flows in a membrane (inlet, permeate and concentrate flows) when the permeate flow is regulated to be kept constant and periodic cleanings are scheduled. From this plot alternative cleaning strategies can be easily evaluated.

5. CONCLUSIONS

This paper has discussed the importance of developing dynamic simulators for designing and operating Reverse Osmosis plants, in terms of optimizing efficiency and designing control software. For this, a library that reproduces the components frequently found in this kind of plants has been presented, showing that it can be used for efficient simulation of RO plants.

6. ACKNOWLEDGEMENTS

This work has been funded by the European Commission within the Sixth Framework Programme (Reference FP6-2004-INCO-MPC-3). We would like to thank SETA, S.L., Empresarios Agrupados, S.A., and the groups working in the EU project “Open-Gain” for feedback and comments.

7. REFERENCES

Alatiqi, I., H. Ettouney, and H. El-Dessouky, ‘Process control in water desalination industry: an overview’. Desalination, 126, 15-32, 1999.

D. Clayton, Reverse Osmosis: the External Treatment of the next Millennium, Analyst 1999.

A. Gambier, A. Krasnik, E. Badreddin, Dynamic Modeling of a Simple Reverse Osmosis Desalination Plant for Advanced Control Purposes, Proceedings of the 2007 American Control Conference. New York, USA, July 11-13, 2007

Gambier, A. and E. Badreddin. Dynamic modelling of MSF plants for automatic control and simulation purposes: a survey. Desalination, 166, 191-204, 2004.

Geislera, P., W. Krummb, T. A. Petersc, Reduction of the energy demand for seawater RO with the pressure exchange system PES. Desalination 135 (2001) 205–210

Jafar, M. and A. Zilouchian, Real-time implementation of a fuzzy logic controller for a seawater RO plant. Proceedings of the 5th World Automation Congress, 13, 31-36, 2002.

Mindler A. and A. Epstein. System identification and control of reverse osmosis desalination. Desalination, 59, 343-379, 1986.

Robertson, M.W., J. C. Watters, P. B. Desphande, J. Z. Assef and I. M. Alatiqi, Model based control for reverse osmosis desalination processes. Desalination, 104, 59-68, 1996.

Seibert, U., G. Vogt, C. Brennig, R. Gebhard, and F. Holz. Autonomous, desalination system concepts for seawater and brackish water in rural areas with renewable energies – potentials, technologies, field experience, socio-technical and socioeconomic impacts – ADIRA. Desalination, 168, 29-37, 2004.

Stover, R.L., Development of a fourth generation energy recovery device. A 'CTO's Notebook', Desalination 165 (2004) 313-321.

Syafiie, S., F. Tadeo, L. Palacin, C. de Prada, Membrane Modeling for Simulation and Control of Reverse Osmosis in Desalination Plants, Control 2008, Cambridge, UK, September 2008.

Wilf, M., The Guidebook to Membrane Desalination Technology, Balaban Desalination Publications, L’Aquila, Italy, 2007.

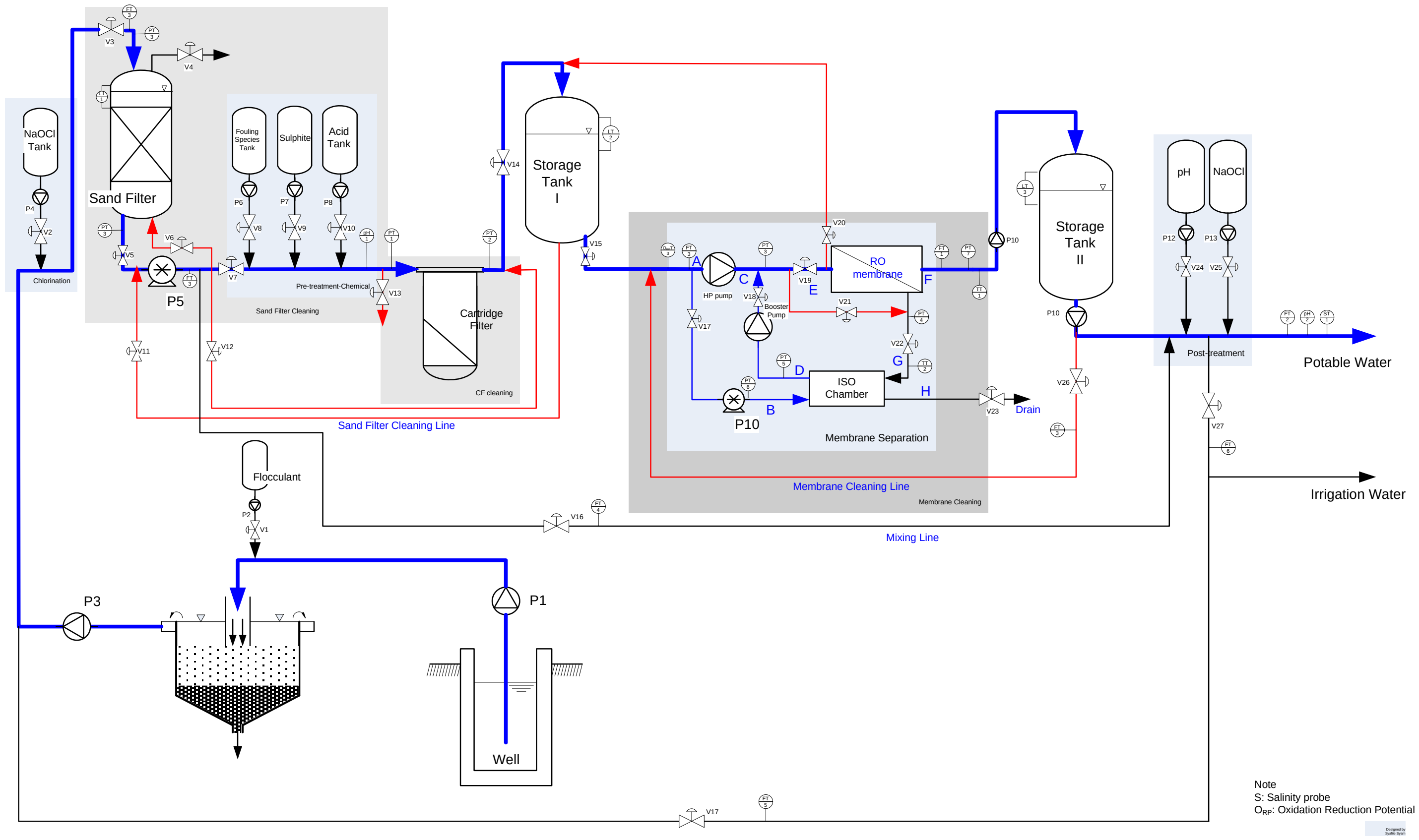


Fig. 2: Layout of a typical RO plant

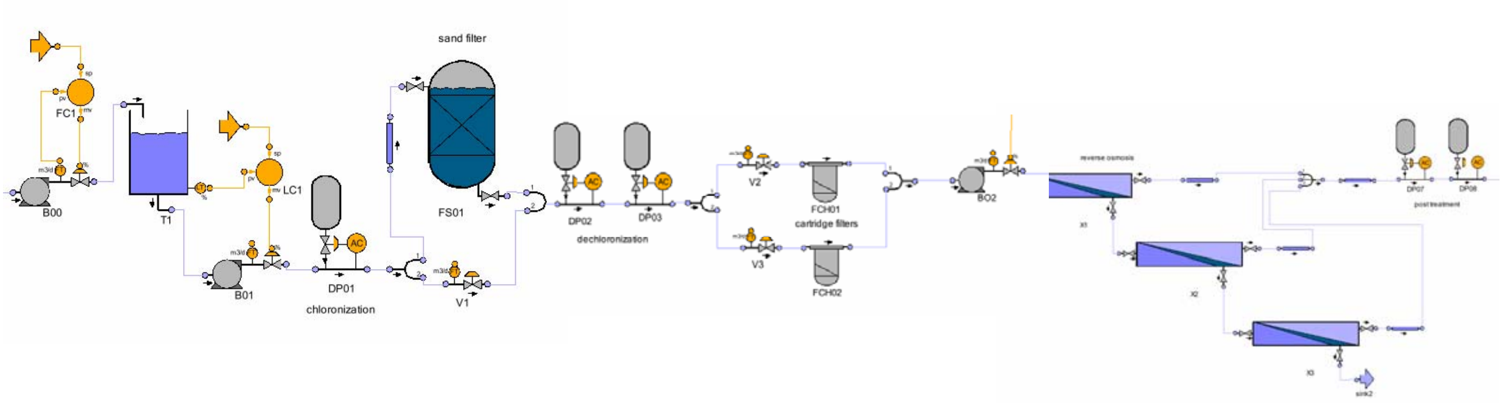


Fig. 3: Graphical simulation of a RO plant

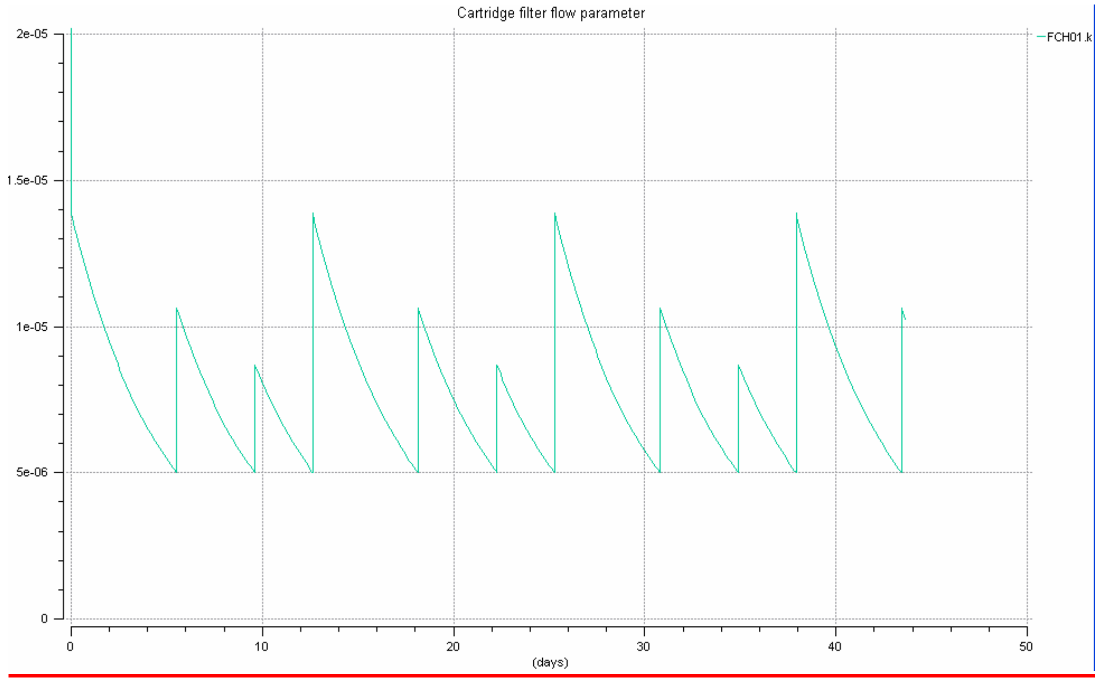


Fig.4: Simulation results for cartridge filter cleaning: permeability evolution with cleanings and replacement for high concentrations of suspended solids.

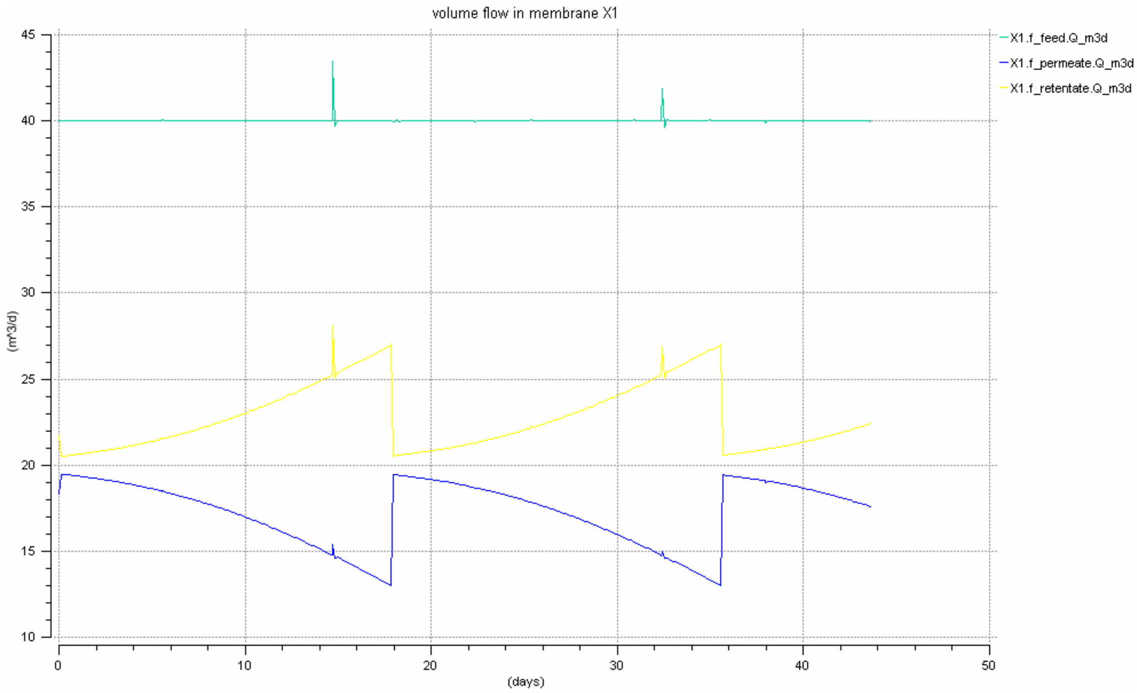


Fig. 5: Simulation results for membrane cleaning: Evolution of flows with cleanings.

USING AN ADVANCED DISCRETE-EVENT SIMULATION FRAMEWORK TO PRODUCTIVE CAPACITY MANAGEMENT OF A CAR-PARTS FACTORY

Adrián M. Aguirre [a], Enrique J. Müller [b], Sebastián E. Seffino [c], Carlos A. Méndez [d]

[a],[b],[c] Industrial Engineering Department (FIQ-UNL), Santiago del Estero 2829, 3000 Santa Fe, Argentina.
[d] INTEC (UNL -CONICET), Güemes 3450, 3000 Santa Fe, Argentina.

[a]aguirreadrianmarcelo@hotmail.com, [b]enriquejmuller@hotmail.com, [c]aseffino@hotmail.com,
[d] cmendez@intec.unl.edu.ar

ABSTRACT

This work presents the development and application of an advanced modelling, simulation and optimization framework focused on the production process of car-parts factory in the Latin-American market. Lying on the concepts of the process-interaction approach, the principal components available in the discrete event simulation environment “SIMUL8” were used to achieve the best representation of the large and complex manufacturing system. Furthermore, advanced SIMUL8’s Visual Logic tools were utilized for modelling specific design and operation features arising in the process under study. The developed tool provides a valuable support system for making and testing operative, tactical and strategic decisions, allowing a quick evaluation of possible sceneries ranging from different operation schemes to potential alternatives of investment. The principal aim of this work is to provide a systematic methodology to validate, evaluate and improve the productive capacity management, enhancing both the process profitability and the degree of customer satisfaction.

Keywords: discrete-event simulation, manufacturing process, SIMUL8, process-interaction approach

1. INTRODUCTION

During the last years Argentina's economic growth was more than twice that of Bolivia, Brazil, Chile and Paraguay, becoming one of the leaders in the region, with major improvements in employment and income distribution and a decline in poverty levels. Within this economic positive context, the automotive industry has been directly affected with a significant increase in car sales and an increasing demand of car-parts. To take advantage of this opportunity, manufacturing plants related to the automotive business are being enforced to improve their process productive capacity. In this work we addressed the particular problem arising in a Company, dedicated to manufacture a high volume and variety of car-parts in Argentina. The major project goals aims at increasing the process productivity and the customer satisfaction.

In order to turn the company more profitable and competitive in a highly demanding market, it was observed the need to rely on effective computational tools to systematically analyze and evaluate the dynamic behaviour of the process, considering different operative schemes and possible alternatives of investment.

Although the use of mixed-integer linear mathematical programming models (MILP) was considered in a preliminary stage of the project, the idea was quickly abandoned due to the lack of proper tools to manage large problem sizes, complex operational logic and different degrees of uncertainty in the problem data. In consequence, discrete events simulation methods were adopted to represent the whole real-world process in an integrated form. In this way, it was possible to clearly identify the problematic situations, providing validated answers to operative, tactical and strategic level requirements involving different sceneries and possible alternatives of investment. Similar applications based on simulation technology to the treatment of operative schemes and alternative of investment in the petrochemical industry can be found in Bacigalupo C.M et al. (2006).

This paper is organized as follows. Section 2 introduces the company under study and the problem to be addressed. The proposed simulation-based methodology is presented in Section 3, describing the steps followed in the model development, the system limitations, the embedded system logic and the user interfaces. Finally, Section 4 introduces some suggested sceneries to give response to the problem challenges and defines the performance indicators to be considered to the study of the system.

2. PROBLEM DEFINITION

2.1. The Company

The main industrial activity of the Company is the manufacturing of a wide variety and a large volume of a basic element of the internal combustion engine, which are commercialized in different markets in a broad range of measures and applications, and under the most rigorous standards of quality. Currently, the company employs more than 50 people to reach a productive capacity of approximately 600.000 pieces monthly. Its present business volume and its integrated manufacturing system allow them to be among the three main car-parts factories in South America**,** reaching to export 52% of its production, almost totally to different Mercosur countries.

The manufacturing process comprises two different production lines and twelve processing stages depending on the main raw materials that are being used, i.e. bars or pipes. The company's policy is to meet the production orders arriving monthly at the time and the form that were previously agreed with the customers. To this end, the process is operated by selecting suitable small and medium production lots.

2.2. Main problems issues

At the present time, the constant raise of car-parts demand has motivated to the general managers of the company to improve the low process productivity associated with three main issues: a) low degree of customer satisfaction because of tardiness, b) large volume of work-in process inventory and (c) work-load unbalance of the manufacturing lines. The combination of these issues generates an inefficient utilization of the limited production resources available to attend the higher demand. Based on the foregoing issues, it were defined a series of challenges directly focused on the actual needs of the company: a) to increase the productivity of manufacturing lines and, b) to respond in more agile and reliable form to the customer requirements.

In summary, the general goal is oriented to optimize the current productive capacity of the company in such a way that most of the deliveries can be met on time and in the right form. In addition, it is desired to be in position of quickly reacting to different commitments in a future with uncertain and variable conditions. Based on the current trend of the economy and the increasing car-parts demand, the company should be able to expand its productive capacity to at least 1.000.000 pieces per month, which constitute the biggest challenge for this work.

2.3. Solution alternatives

Different available solution alternatives were evaluated to cope with this problem. The use of an analytical method based on linear mathematical formulation tools (MILP models), which are broadly used to production optimization, was avoided because of the inherent computational and modelling limitations of these techniques. Also, the use of simple heuristic procedures was considered but they failed to handle the high complexity of the problem. Although these two approaches may be valid to generate optimal solutions for the system, they are not capable of effectively managing the random nature of many process variables (e.g. processing time distributions, material movement time distributions, etc.), the large number of resources involved and the complex operational rules to be considered. In order to overcome the limitations previously stated, it was decided to make use of a modern and user-friendly system analysis tool by developing an advanced computational process model lying on the notion of discrete event system simulation (Barcelo 1996; Banks et al. 2004; Koshnevis 1994)

2.4. Advantages of simulation-based methods

The main advantage of the proposed computer-aided methodology is that it permits to systematically reproduce the complex company process in an abstract and integrated form, visualizing the dynamic behaviour of its constitutive elements over time.

Production facilities (work centers), storage facilities (tails of waits, deposits of raw materials, intermediate storages and process storages), transports and materials movement devices (vehicles, bridge cranes, etc), external factors (variability in the arrival of the raw materials and demands), internal and unforeseen factors (changes in policies and operational rules, breaks, stops and maintenances of processing units in the production lines), and resources of the system (e.g. workers), were analyzed and modelled in detail. As result, a basic model was generated to discern the principal weaknesses and bottlenecks of the process, but it was also useful later for making decisions to enhance the current process performance.

3. THE PROPOSED SIMULATION-BASED FRAMEWORK

In order to formulate a precise computer-aided representation to the real-world manufacturing system described above, it was decided to make use of the simulation, visualization and analysis tool set provided in the discrete events simulation environment "SIMUL8" (Shalliker et al. 2002; Haige et al. 2001; Mc Gregor et al. 2004; SIMUL8 2004) . The main characteristics of the proposed basic model are described in detail below.

3.1. Principal model features

- Raw Materials: depending on the raw material being used in the process (pipes or bars), two alternative manufacturing lines were modeled. The main raw material characteristics are detailed in the Table 1.

Table 1: Raw Materials Types

Raw material	Outside diameter [mm]	Length [meters]
bars	22 – 60	5.5 – 6
pipes	16 - 28.6	6

- Product Family: three main final product types were identified according to the principal characteristics of the process (see Table 2).

Table 2: Final Product Types

Product family	Characteristics
Small car-parts	Outside diameter <25,4[mm]
Medium car-parts	Outside diameter <38,1[mm]
Big car-parts	Outside diameter <60,3[mm]

- Manufacturing process stages: The production sequence for every car-part family depends on the raw material being used. The general manufacturing stages are listed in Table 3 whereas the specific product recipes for every car-part family and raw material are shown in Table 4 and Figure 1. In turn, the entire simulation model developed in the SIMUL8 environment can be found in Figure 2. Additionally, a partial size view of the in-progress SIMUL8 model can be seen in Figure 3.

Table 3: General Manufacturing Process Stages

Process stage	Description	Available units
0	Raw material reception	-
1	Bars cutting process	3 (cutting machines)
2	Re-cooking process	3 (ovens)
3	Soaping process	7 (vats)
4	Pressing process	6 (press)
5	Uncovering process	2 (press)
6	Rectifying process	2 (machines)
7	Cutting and beveling process	4 (machines)
8	Thermal treatment process	12 (vats)
9	Final rectifying process	4 (machines)
10	Pipe cutting process	1 (cutting machine)
11	Stamping process	2 (stamping machines)
12	Final control	(manual process)

Table 4: Process Sequence for Each Product Family

Family	Raw material	Production sequence (stages)
Small car-parts	Bars	0-1-2-3-4-5-2-3-4-6-7-11-8-9-12
Medium car-parts	Bars	0-1-2-3-4-5-2-3-4-6-7-8-9-12
Big car-parts	Bars	0-1-2-3-4-5-2-3-4-6-7-8-9-12
Small car-parts	Pipes	0-10-11-6-8-9-12
Medium car-parts	Pipes	0-10-11-6-8-9-12

- Lot-sizing: Every order of product is divided into a number of batches for its processing and movement inside the factory. The maximum allowable batch size is defined in Table 5. This limitation comes mainly from the maximum feasible processing load and the available capacity of material handling resources.

Table 5: Maximum Batch Size for Each Product Type

Small car-part	Medium car-part	Big car-part
2500	1500	500

- Shift timetable: Three 8-hour work shifts are contemplated from Monday to Saturday. Also, an additional shift for Sunday is defined. The same ones are detailed in Table 6.

Table 6: Shift Timetable

Shift	Start time [hrs]	End time [hrs]	Days
Morning	6:00	14:00	Mon - Sat
Afternoon	14:00	22:00	Mon - Sat
Night	22:00	6:00	Mon - Fri
Sunday	6:00	14:00	Sun

- Resources of the System: Available resources for storage in transit, temporary storage and movement of materials are detailed in Table 7.

Table 7: Available Production Resources in the Factory

Resource name	Description	Quantity available
Baskets	Resource for the storage in process	150
Pots	Resource for the storage in process	50
Soaping baskets	Resource for the storage in process	6
Cages	Resource for the storage in process	50
Hydraulic elevators	Manual vehicle for materials handling	5
Bridge crane	Device for materials handling	1
Manual Vehicles	Manual vehicle for materials handling	2
Hoist	Vehicle for materials handling	1
Conveyors	Device for materials handling	10

- Available manpower: The workers were modeled as an additional manufacturing resource in the production system with a dynamic availability over time. The allocation of workers to processing stages depends directly on the work shift and it is detailed in Table 8. In addition, programmed and non-programmed breaks and absenteeism can be easily incorporated in the simulation model.

Tabla 8: Dynamic Manpower Availability in the Factory

Resource name (Workers)	Processing stages	Worker availability for each work shift			
		Morning	Afternoon	Night	Sunday
Cut worker of bars	1	1	1	1	0
Re-cook worker	2-3	1	1	1	1
Maintenance worker	all	2	2	2	2
Press worker	4-5	4	4	3	3
Rectified, cut and beveled worker	6-7	3	4	3	0
Thermal treatment worker	8	4	4	4	0
Final rectified worker	9	4	4	0	0
Final control worker	12	5	5	0	0
Worker of pipes	10-11-6-8-9	1	1	0	0

- Data modeling: The processing time of the principal units was modeled using a normal probability distribution, which depends on the product type, the operation stage and the characteristic of the raw material that is being used (i.e. outside diameter). The validated parameters of the distribution were obtained after conducting the corresponding statistics studies. Likewise, two different setup times were modeled using deterministic values. The first one depends on the changes in product type and the current operation process. The second one was modeled for changes in process operations, i.e. for multipurpose units. These conditions were carefully programmed by using SIMUL8's Visual Logic functions (VL) for each distribution. Also, uniform time distributions for materials movements and loading and unloading of materials in process were employed.

3.2. Model development and operative rules

Through the development of the simulation model, it were identified all the manufacturing restrictions and operational rules to be considered. These type of restrictions can be caused by operative intrinsic factors (e.g. maximum batch sizes for every car-part family, available manpower in each work shift, process operations and manufacturing stages allowed in each work center, etc.) as well as external factors (e.g. raw material shortfalls, demand variability, etc.).

These features were directly modelled by using the Visual Logic function (VL) available in the SIMUL8 software, based on the occurrence of different system events. An illustrative example for the embedded logic behind the order priority definition is shown Figure 4. Other actual manufacturing restrictions and operating rules are described below.

1. Production batch sizes are based on the maximum capacity defined for every product type. The selected size is maintained through the entire production process, comprising processing stages and material movements). The material movement task is carried out for every batch. This activity requires a resource for storage in transit (e.g. one basket), the corresponding resource for material movement (e.g. an hydraulic elevator) and the allocated manpower (e.g. one of the press workers).
2. The material movement activities have higher priority than any other processing activity.
3. Processing units are modelled as unary resources, i.e. batches are processed one by one in every work center, except in certain centers like ovens. In these manufacturing resources, two batches must be processed simultaneously. Each oven can perform two operations (1st re-cooked and 2nd re-cooked). In every work load, it can be mixed batches of different product types but not batches of different operations.
4. Some facilities can perform several operations on different product types (multipurpose units). For example, some presses are able to carry out three operations for every product type whereas the ovens can perform two operations (1st re-cooked and 2nd re-cooked). Each kind of operation has a different processing time and setup time distributions. In consequence, tasks allocations for every product type at every work center are carefully managed in such a way that setup times can be minimized.
5. Raw materials (pipes and bars) are classified into several subtypes depending on the outside diameter. In order produce a particular order, it must be guaranteed the use of those subtypes (pipes or bars) in which the outside diameter is equal or larger than the diameter of the order. In case that at some time there is no availability of raw materials satisfying the outside diameter, the beginning of order must delayed up to the reception of the corresponding supply of material.
6. In every manufacturing stage, the processing of production orders is arranged depending on the predefined commercial priority.

7. In the case that all resources for material storage were totally busy, the model does not allow the assignment of resources to certain work centers in order to avoid any resource blockade.

In the same way, a series of dialogs and menus were developed by using Visual Basic tools in order to provide an interactive and user-friendly model interface. These tools allow the user to enter all the problem data and also to propose changes to the design and operative conditions of the production system. Moreover, it provides an effective way to analyze and study model results during the simulation run (see Figures 5 to 9).

Once the problem data was loaded in the simulator interfaces and the restrictions and operational rules were correctly programmed, it was possible to perform the validation of the simulation model in order to guarantee that the model imitates the current process conditions. Also, it was carefully observed if the model was slightly sensitive to superfluous changes in the system.

To do this, it was defined and analyzed a set of measures and indicators of performance and effectiveness in order to corroborate the correct computational representation of the real system. Total production volume, fulfilment of production goals, average amount of in-process inventory in every manufacturing stage were some of the system output variables that were analyzed in detail.

The study of the dynamic behaviour of certain system resources, such as machines, storage resources, workers and in-process inventory levels in each stage, was carried out using different SIMUL8 visual tools oriented to system output analysis (see Figure 8 and 9).

On the basis of the simulation results, it was possible to conclude that the development of the analyzed and validated basic model was useful in a preliminary project stage to identify the possible weak points in the manufacturing lines and to visualize the complex dynamic of global process.

This basic model is the principal tool for the study and analysis of alternative system solutions. In addition, it represents a suitable model to evaluate future decisions that could improve the proposed criterion as well as to propose better solutions that help to increase the current process performance.

4. CONDUCTING SIMULATION STUDIES

After identifying the major operational issues in the process, it was proposed a series of changes related to design and operative features, which a priori could increase the productive capacity of the company and fix the weak points of the process.

To do this, alternative sceneries are to be generated, identifying the values of the factors that are considered to be more relevant. Some of them are explicitly defined below.

- Increasing the available capacity of the resources associated with the material movement.
- Increasing the existing manufacturing resources in the likely bottlenecks.
- Reassignment and dynamic re-scheduling of available workers and resources of the system.
- Increasing the number of machines in the principal process stages.
- Modification of some basic operative rules:
- Order sequencing in shared manufacturing resources (e.g. minimizing the number of setup times for work center, pre-assignment of tasks and products to be processing in every work center, modification in the processing sequence, etc.).

4.1. Measures of Effectiveness

Our primary measures of effectiveness and performances are the following ones:

- Production volume per type of product (car-parts/Week or Pieces/Month).
- Production volume in every work center (car-parts/Week or Pieces/Month).
- Percentages of utilization of workers, work centers and resources of the system.
- Average in-process inventory in every stage

5. CONCLUSIONS

This work presented the major steps followed for the development and the application of an advanced modelling, simulation and optimization tool to productive capacity management of a factory of a basic element of the internal combustion engine. It allows to easily represent and validate the existing operative schemes in the company in order to determine the weak points in the manufacturing lines that need to be improved. In addition, the developed tool can be used to analyse, test and predict the dynamic behaviour of the system after introducing modifications in the production schemes and/or possible alternatives of investment.

ACKNOWLEDGMENTS

The authors are thankful for financial support from FONCYT-ANPCyT under Grant PICT-01837, from CONICET under Grant PIP-5729 and from Universidad Nacional del Litoral under CAI+D 003-13.

REFERENCES

Bacigalupo, C.M, Gratti, C.A, Martinez, A.O, Franzece, L.A, Fiorini, M.M, Paz, D.P., 2006. *Simulación computacional de la cadena logística de la refinería San Lorenzo*, Revista del Instituto Argentino del Petróleo. Available from: http://www.paragontech.com.ar [accessed 1 March 2008].

Barceló, J., 1996. *Simulación de Sistemas discretos.* España., Isdefe.

Banks, J., Carson, J, Nelson, B y Nicol, D., 2004. *Discrete event system simulation,* Fourth edition. USA, Prentice-Hall.

Koshnevis, B., 1994. *Discrete systems simulation*. USA, McGraw-Hill.

Law, M. and Kelton, W., 1991. *Simulation modeling and analysis*. USA, McGraw-Hill.

Shalliker, J., Ricketts, C., 2002. *An Introduction to SIMUL8*. Release nine. School of Mathematics and Statistics, University of Plymouth.

SIMUL8: Manual and User Guide. 2004. Visual Thinking International: Mississauga, Ontario Canada.

Haige, J.W. and Paige, K.N., 2001. *Learning SIMUL8: The Complete Guide*. Plain Vu: Bellingham, WA.

Mc Gregor, D.W, Cain, M.J., 2004. *An introduction to SIMUL8*. Department of Management, College of Business and Economics, University of Canterbury, New Zealand.

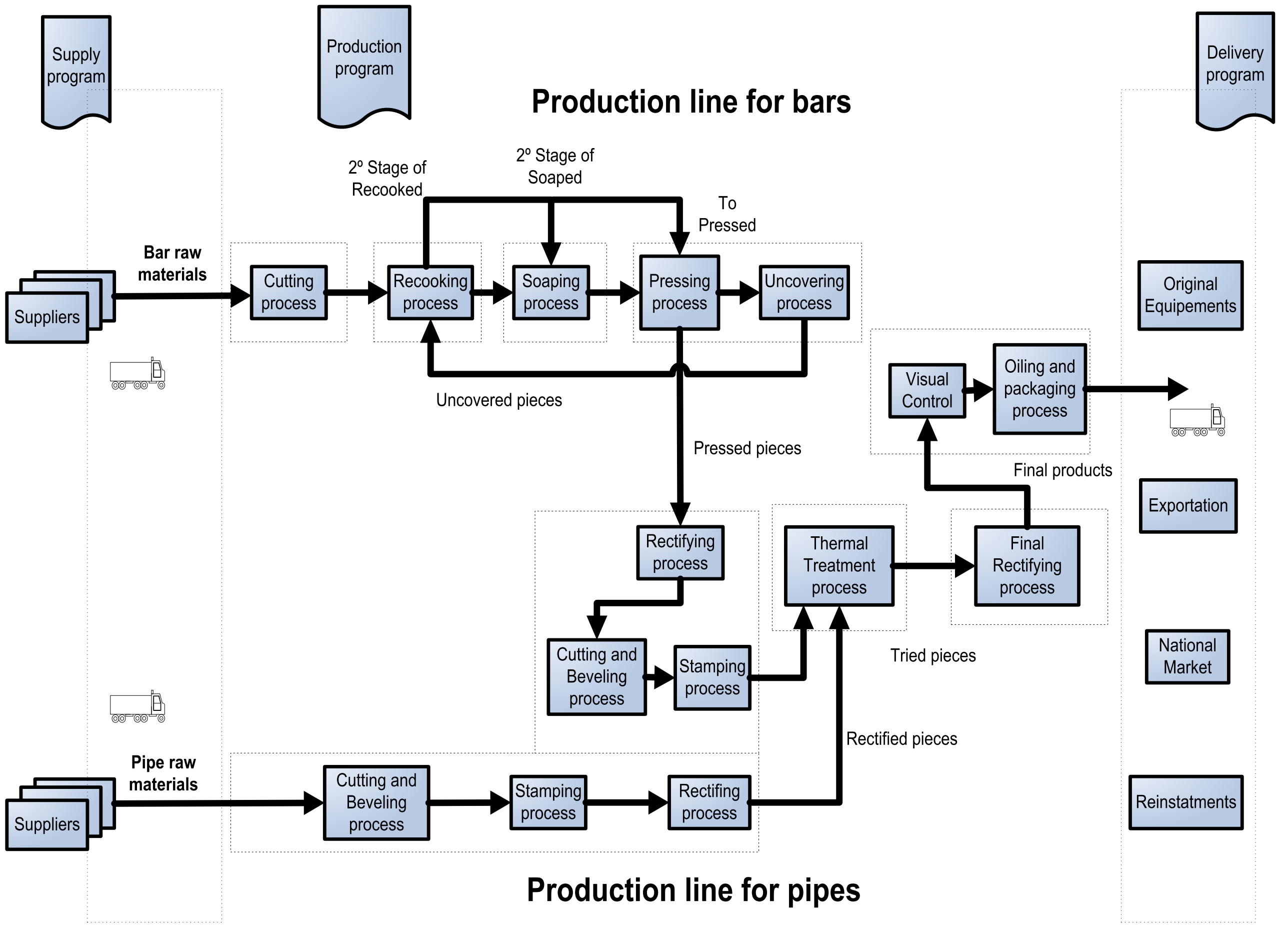


Figure 1. Global Manufacturing System of the Company

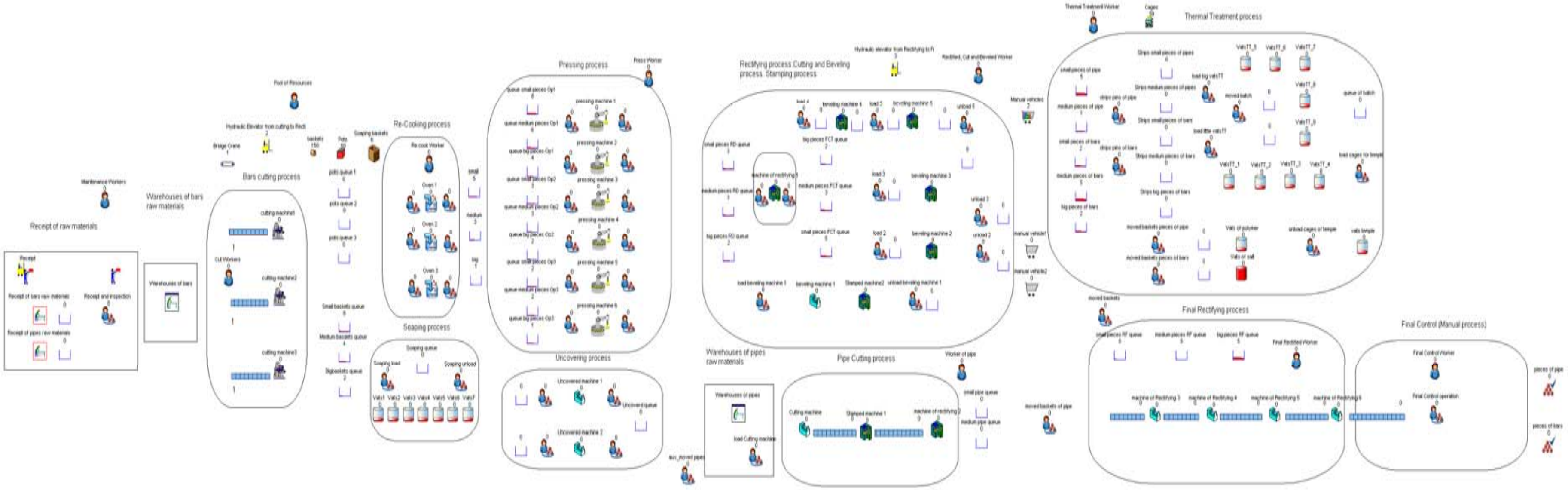


Figure 2. Computer-based Representation of the Entire Manufacturing Process Model Generated in the SIMUL8 Environment

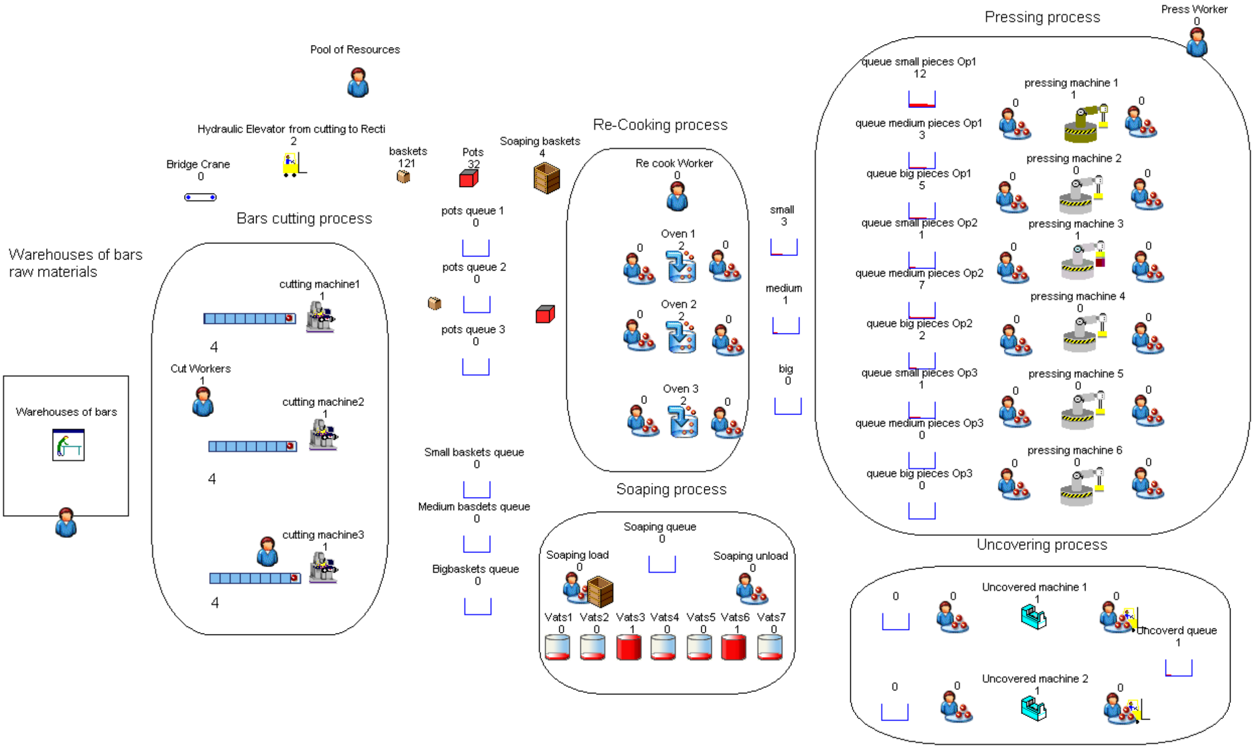
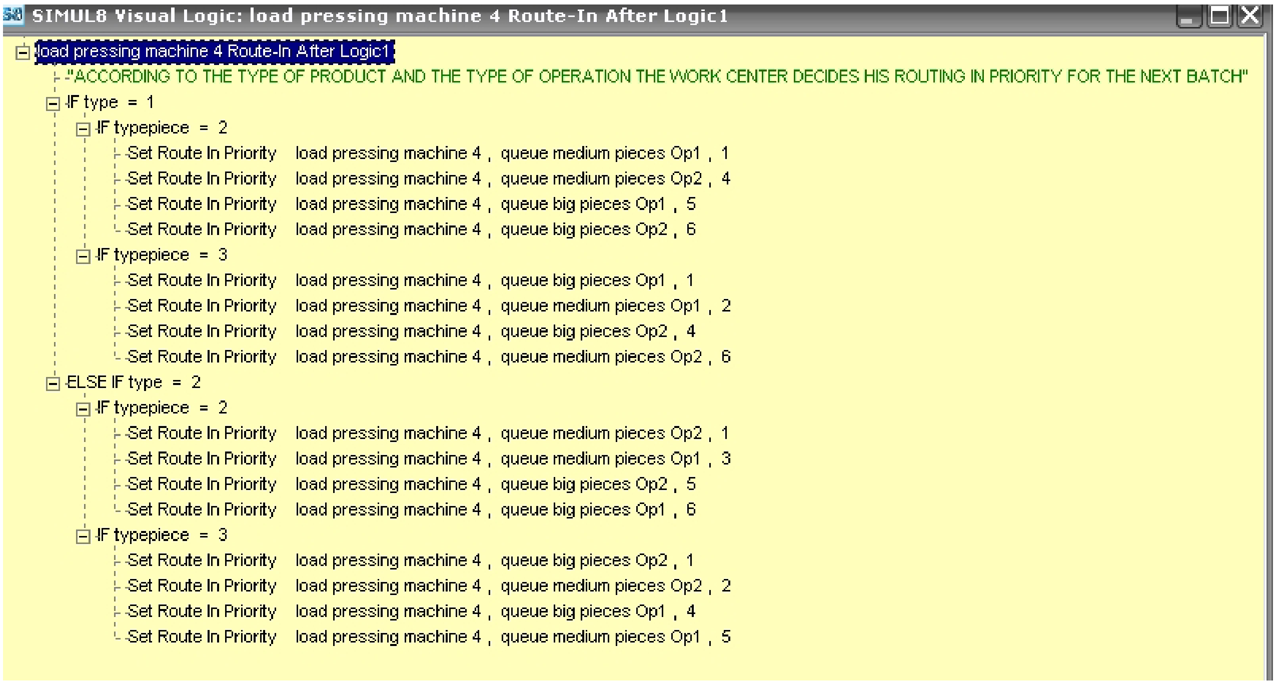


Figure 3. Partial Size View of the In-progress SIMUL8 Model


SIMUL8 Visual Logic: load pressing machine 4 Route-In After Logic1

```
load pressing machine 4 Route-In After Logic1
  'ACCORDING TO THE TYPE OF PRODUCT AND THE TYPE OF OPERATION THE WORK CENTER DECIDES HIS ROUTING IN PRIORITY FOR THE NEXT BATCH"
  IF type  =  1
    IF typepiece  =  2
      Set Route In Priority   load pressing machine 4 ,  queue medium pieces Op1 ,  1
      Set Route In Priority   load pressing machine 4 ,  queue medium pieces Op2 ,  4
      Set Route In Priority   load pressing machine 4 ,  queue big pieces Op1 ,  5
      Set Route In Priority   load pressing machine 4 ,  queue big pieces Op2 ,  6
    IF typepiece  =  3
      Set Route In Priority   load pressing machine 4 ,  queue big pieces Op1 ,  1
      Set Route In Priority   load pressing machine 4 ,  queue medium pieces Op1 ,  2
      Set Route In Priority   load pressing machine 4 ,  queue big pieces Op2 ,  4
      Set Route In Priority   load pressing machine 4 ,  queue medium pieces Op2 ,  6
  ELSE IF type  =  2
    IF typepiece  =  2
      Set Route In Priority   load pressing machine 4 ,  queue medium pieces Op2 ,  1
      Set Route In Priority   load pressing machine 4 ,  queue medium pieces Op1 ,  3
      Set Route In Priority   load pressing machine 4 ,  queue big pieces Op2 ,  5
      Set Route In Priority   load pressing machine 4 ,  queue big pieces Op1 ,  6
    IF typepiece  =  3
      Set Route In Priority   load pressing machine 4 ,  queue big pieces Op2 ,  1
      Set Route In Priority   load pressing machine 4 ,  queue medium pieces Op2 ,  2
      Set Route In Priority   load pressing machine 4 ,  queue big pieces Op1 ,  4
      Set Route In Priority   load pressing machine 4 ,  queue medium pieces Op1 ,  5
```

Figure 4. Representation of Operative Rules Through SIMUL8's Visual Logic Programming

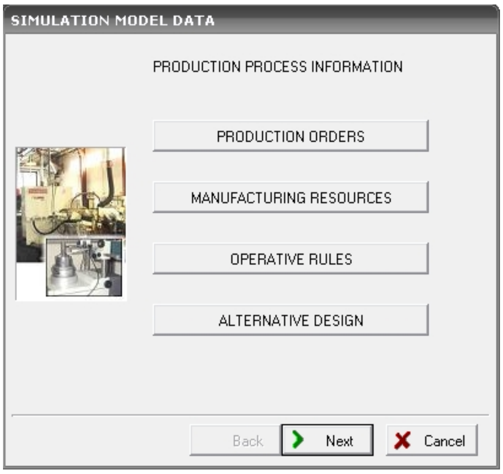
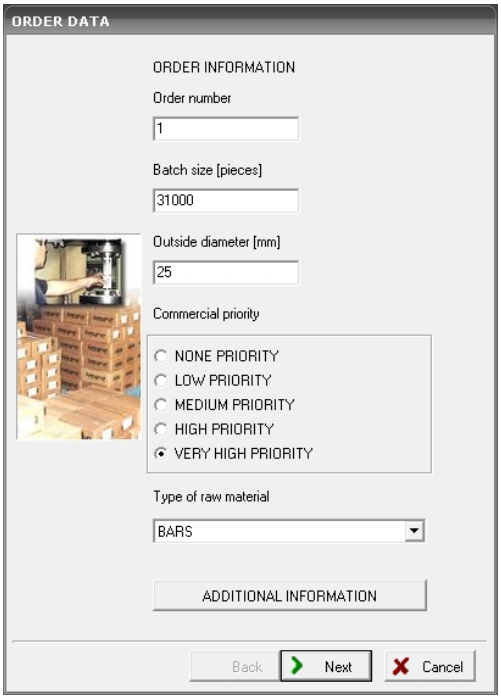


Figure 5: Principal Menu



Order number: 1

Batch size [pieces]: 31000

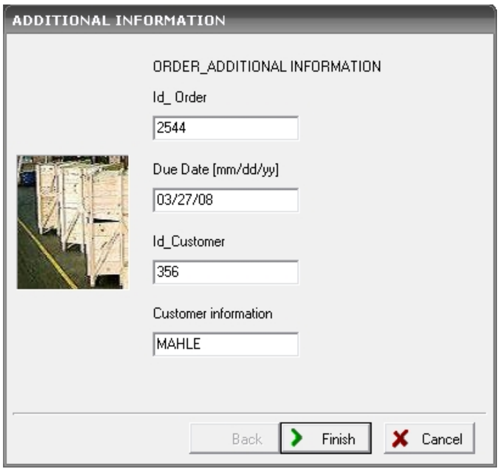
Outside diameter [mm]: 25

Commercial priority

○ NONE PRIORITY
○ LOW PRIORITY
○ MEDIUM PRIORITY
○ HIGH PRIORITY
● VERY HIGH PRIORITY

Type of raw material: BARS

Figure 6: Interface Order Data



Id_ Order: 2544

Due Date [mm/dd/yy]: 03/27/08

Id_Customer: 356

Customer information: MAHLE

Figure 7: Interface for Detailed Order Information

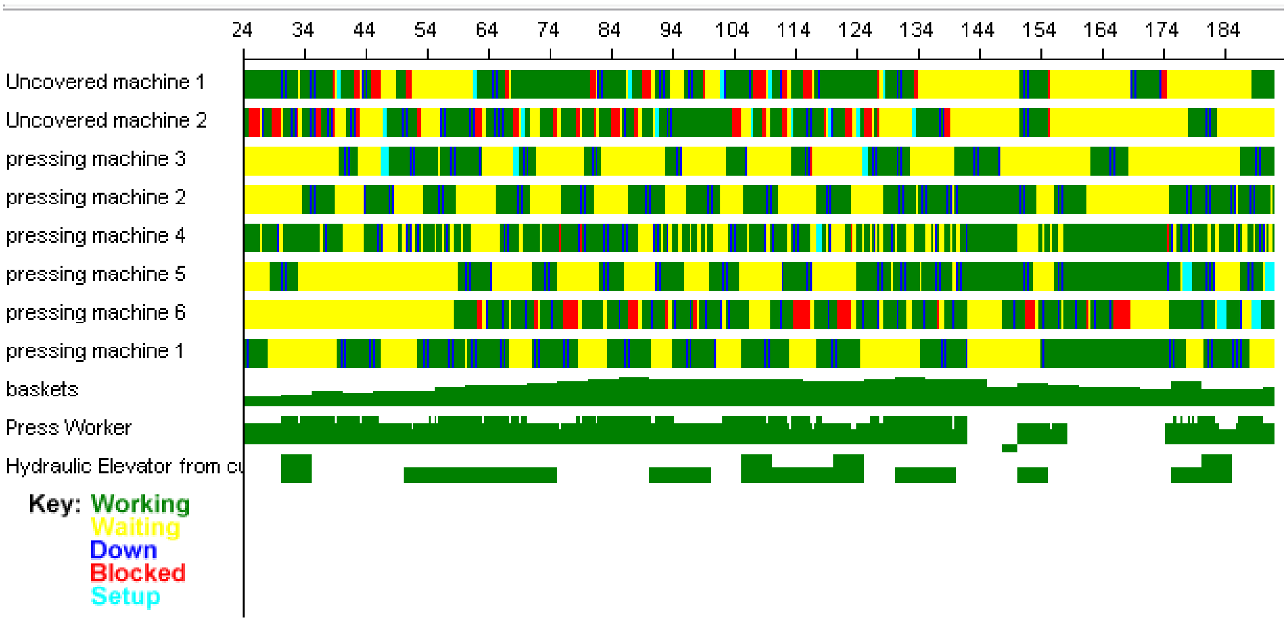


Figure 8: Dynamic Evolution of Manufacturing Resource Utilization Along the Simulation

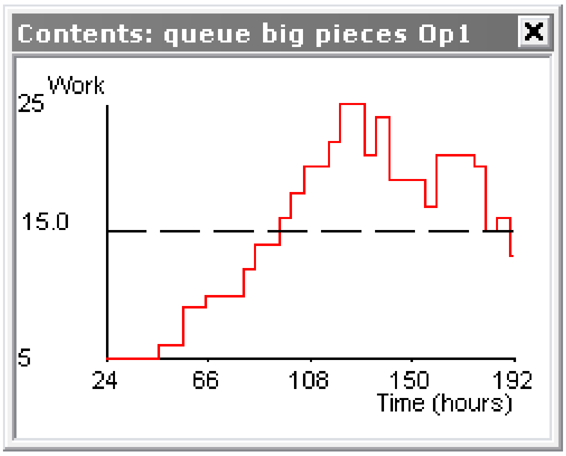


Figure 9: Dynamic Evolution of the In-process Inventory Level of “queue big pieces Op1” in the Pressing Stage.

CLASSIC AND ADVANCED MODELS FOR CONTROLLING HVAC SYSTEMS IN A UNIVERSITY BUILDING

Eladio Sanz[(a)], Belén Pérez-Lancho[(a)], Pastora Vega[(b)] , Mario Francisco[(b)]

[(a)] Dept. Informática y Automática. Fac. Ciencias. Univ. Salamanca.
Plaza de la Merced, s/n. 37008 Salamanca
[(b)] Dept. Informática y Automática. E.T.S. Ingeniería Industrial. Univ. Salamanca.
C/ Fernando Ballesteros, s/n. 37700 Béjar (Salamanca)

[(a)]{esanz, lancho}@usal.es, [(b)]{pvega, mfs}@usal.es

ABSTRACT

In this work a systematic procedure for the modelling and simulation of an HVAC (Heat Ventilating and Air Conditioning) thermal system is reported. The dynamic behaviour of each element, not only on its own but also as part of the whole process, is analysed and tested. The basic principles of energy conservation and heat transmission are applied to model the various processes that usually occur in thermal installations. An experimental building was used to evaluate the proposed energy control; it was also used to identify methods for modelling both particular elements and the whole system. Following validation of the model, new control strategies for the process, with classic and advanced algorithms and procedures, are proposed with a view to achieving both comfort and important energy savings.

Keywords: Simulation, HVAC System (Heat Ventilating and Air Conditioning), MPC (Model Predictive Control).

1. INTRODUCTION

Energy consumption in domestic and public buildings accounts for almost 30% of the total energy consumption of the European Union. CO_2 pollution due to maintaining comfortable temperatures inside buildings represents a similar percentage. Considerable efforts are now under way to reduce energy waste in thermal installations; such efforts involve several aspects, such as creating new elements and materials and research into new control and management systems. The two main goals underlying these endeavours are comfort and energy savings. Naturally, even small energy savings in an individual module will afford appreciable benefits when the strategies proposed here are applied in larger scale.

The relationship between a building's architecture and its thermal engineering can be found in (Kuehn et al. 1998). In (García-Sanz 1997) the author reports an analogy between electrical circuits and thermal systems and uses it to model room temperatures. In previous works (Mathews et al. 1999; Riederer 2002; Stec et al. 2005) the authors applied different methods to analyse and study the problem in specific systems. In this sense, a systematic study, both theoretical and experimental, can be found in (Liao et al. 2005). Important contributions in this field include a modular thermal model with heat transfer between zones, affording a system of stochastic differential equations with statistical estimation of their parameters (Andersen et al. 2000), and a recent work with a model predictive control strategy (Shui et al. 2006).

In the present work a systematic procedure for the modelling and simulation of a real HVAC thermal system is reported. Based on real data from an experimental building, identification procedures are generated in order to calculate and estimate the unknown parameters of the process. The model is then validated, taking into account the same experimental data collected along the general system function. New control strategies for the process are proposed, using classic and advanced algorithms and the procedures are tested in order to achieve both comfort and important energy savings.

2. PROCESS DESCRIPTION

The three main elements of the HVAC system (Fig. 1) modelled here can be classified thus:

- Energy-producing units (Boiler and Cooler)
- Energy-exchange units (Air Handling Unit and radiators)
- Zones/Rooms

In these types of system, it is common that there will be coupled signals and fluid flow feedbacks (of air and water) that make analysis of the dynamics of the system both tedious and complex. This situation produces non-linear equations, time delays, and non-homogenous characteristic dynamics. Additionally, there are further elements and signals which must be taken in to account:

- Fluid transmission elements (pipelines, ducts)

- Flow control elements (thermostats, three-way-valve electronic regulators, variable air volumes)
- Classic and advanced controllers (PID, Model Predictive Control)
- Internal and external disturbances (external temperature, solar radiation, occupancy and lighting).

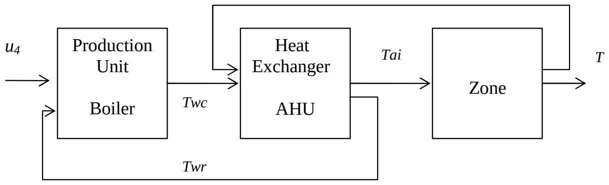


Figure 1: General scheme of an HVAC system

Many of these elements have been modelled in different ways; some of the models are fairly simple and others are somewhat more convoluted and complex. Choice of the most appropriate model depends on its intended application (design, control, or education). For instance, models for the indoor zones of buildings may be described by means of first-order linear equations, although it is also possible to use non-linear models with high-order dynamics. Under general assumptions the governing equations of the main elements are:

<u>Zone model</u>

$$\begin{aligned} C_R \frac{dT}{dt} &= -\rho_a c_a (T - T_{ai}) u_1 q_R - (UA)_v (T - T_{ex}) - \\ &- (UA)_{pi} (T - T_p) + Q_p \\ C_p \frac{dT_p}{dt} &= -(UA)_{pe} (T_p - T_{ex}) + (UA)_{pi} (T - T_p) \end{aligned} \tag{1}$$

<u>Heat Exchanger (Air Handling Unit - AHU)</u>

$$\begin{aligned} C_a \frac{dT_{ai}}{dt} &= -\rho_a c_a q_a T_{ai} + \\ &+ \rho_a c_a (u_3 q_a T_{ex} + (1 - u_3) q_a T) + (UA)_{AHU} (T_{wr} - T_{ai}) \\ C_w \frac{dT_{wr}}{dt} &= -\rho_w c_w q_w T_{wr} + \rho_w c_w (u_2 q_w T_{wc} + \\ &+ (1 - u_2) q_w T_{wr}) - (UA)_{AHU} (T_{wr} - T_{ai}) \end{aligned} \tag{2}$$

<u>Production Unit (Boiler)</u>

$$\begin{aligned} C_c \frac{dT_{wc}}{dt} &= \rho_w c_w u_2 q_w T_{wr} - \rho_w c_w u_2 q_w T_{wc} - \\ &- (UA)_c (T_{wc} - T) + \eta P_N u_4 \end{aligned} \tag{3}$$

<u>Nomenclature</u>

C	*overall thermal capacitances: zone (C_R), walls (C_p), air in the AHU (C_a), water in the AHU (C_w) and boiler (C_c)*
c	*specific heat of air (c_a) or water (c_w)*
ρ	*air density (ρ_a) or water density (ρ_w)*
q	*volume flow rates: supply air in the zone (q_R) or in the AHU (q_a) and water in the AHU (q_w)*
Q_p	*heat gains from occupants and lighting*
(UA)	*overall heat transfers: windows $(UA)_v$, external walls $(UA)_{pe}$, internal walls $(UA)_{pi}$ and boiler $(UA)_c$*
$(UA)_{AHU}$	*overall transmittance area factor of the AHU*
T	*zone measured temperature*
T_{ai}	*supply air temperature*
T_{ex}	*outside temperature*
T_p	*inner wall temperature*
T_{wc}	*water supply temperature in 3- way valve*
T_{wr}	*water return temperature in 3- way valve*
u_1	*damper zone control*
u_2	*three way valve position*
u_3	*outside air damper position*
u_4	*gas/fuel flow rate*
η	*boiler efficiency*
P_N	*nominal power*

The models obtained here follow a block-oriented approach; *Matlab/Simulink* was selected as the simulation toolbox used to study, analyse, and design partial and general simulation scenarios. A new library, known as HVAC, was built in order to simplify model generation, and also for use as a tool to interpret the simulation results.

The availability of real data, coming from a university building where an HVAC system is installed, enabled us to carry out tests to validate the partial and global models created; procedures for the identification of processes were also generated with these data. Once the HVAC model results have been confirmed by experimental data, new and improved control strategies could be designed.

3. REAL PROCESS AND MODELLING

As a demonstration of both the simulated process and the real one, figure 2 shows the block diagram, built n *MATLAB/Simulink* modules of a large space, the reading room of a University building which has been selected in this experiment. Input/output data collected from the real system that can be shown are: input heat flow rate, supply input air temperature, internal and external temperature and periodical starting/shutdown signals.

A functional block representing the zone model is included in the same file. In order to compare correctly the results of both systems, the real process and its model, the same external real signals (control and disturbances) act over the model.

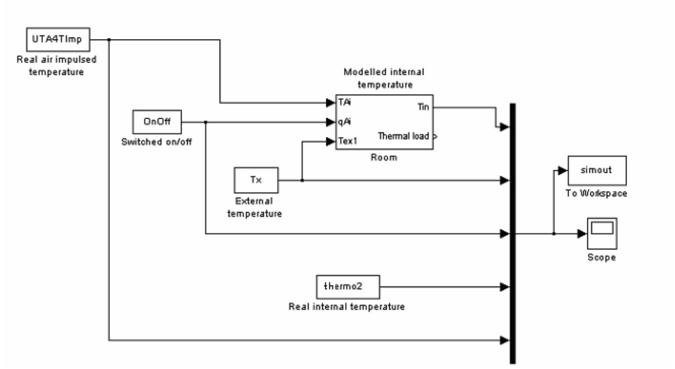


Figure 2: Real and modelled zone thermal system

The graphic results for room temperature during a given period of time are shown in figure 3. It is easy to observe a good fit between the simulated and experimental results. In this case, it was possible to achieve identification with a second-order system for indoor temperature, and a first-order model for air flow. Supply air temperature was used as the control variable, and outside temperature was taken as a disturbance input.

As shown in figure 3, it was not possible to achieve either the comfort or the energy expenditure goals during the period of time under consideration in the real system; again, a strong overheating and a rather poor energy performance can be observed from start to end. Basically, the reason was a fault in design, together with poor control strategy.

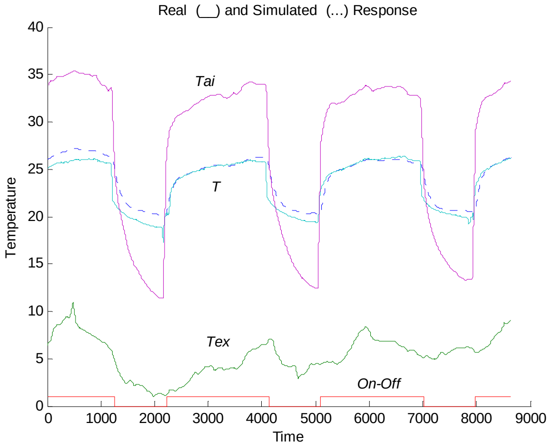


Figure 3: Real and modelled zone thermal results

However, by recording all the available signals during this period of time, allows identify the process, not only for the studied zone but also for the other dynamic elements in the system. This systematic method furnished partial or global models that were useful for testing and simulating future control strategies.

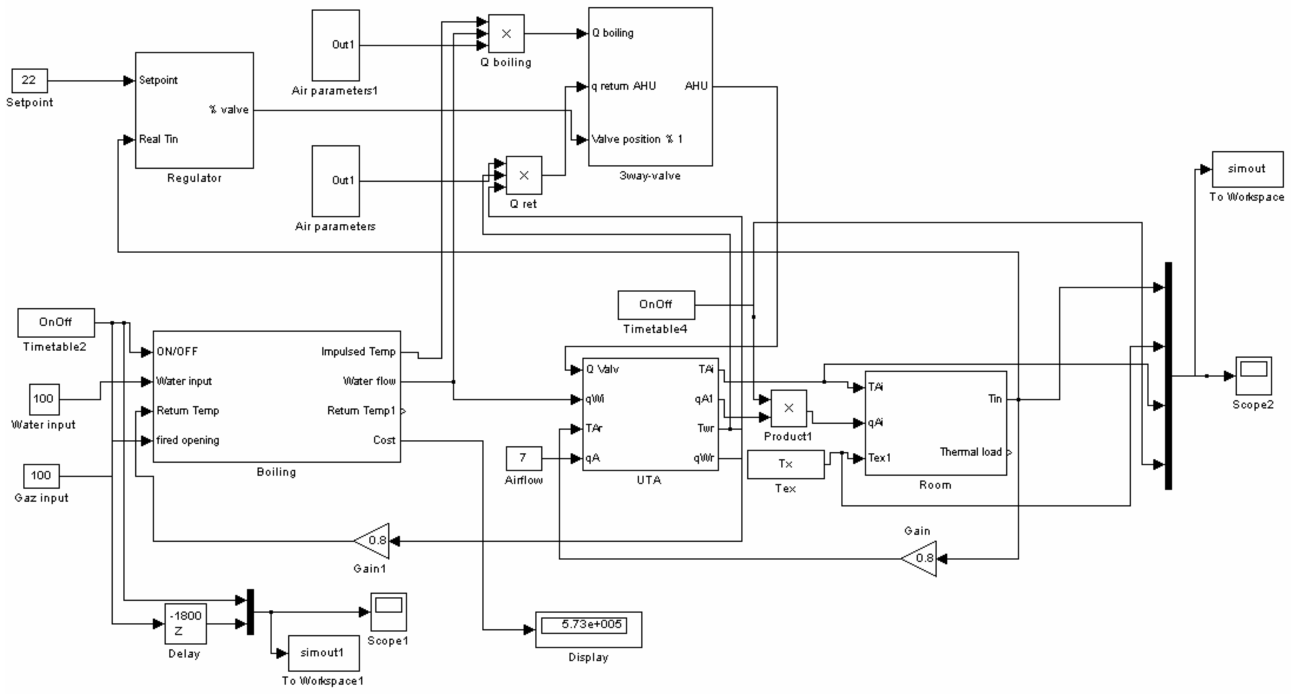


Figure 4: HVAC model system with 3-way valve regulator

4. CONTROL STRATEGIES

Once the model has been validated, alternative control strategies have been designed in order to improve performance having as objectives comfort and energy savings. Different subsystems representing the functional performance of physical elements have been taken into account. The unit production, the AHU, and the zone models are presented as complex blocks in *Simulink*. Signals, external data and input/output ports allow the interconnection of the main elements.

Special attention has been paid to the design of control units. In order to approximate this model to the practical situation in a building a three way valve has been modelled as the only actuator in the process.

Usually, the process control is carried out either using an ON/OFF controller or a conventional PID. Furthermore, most building installations use just a P controller due to the fact that a PI controller could seriously damage the valve and otherwise and small stationary errors in internal temperature are allowable in such large zones with great occupation rates.

Such a simple control which acts over the three way valve, in the sense of correctly mixing heat water flows from boiler and return ducts, is the base for interchanging energy with the air in the AHU which is the nucleus of HVAC system. The whole system *Simulink* model is presented in Fig. 4.

Figure 5 shows the results when a P controller (with a setpoint of 22ºC and a gain of 50) and a start/shutdown system were implemented. Small errors can be observed in the temperature variable due to the lack of integral action. Moreover the evolution of the supply air temperature is presented. There can be seen a small gap between both temperatures, which is enough to achieve the zone comfort, that is usually considered an interval from 20º to 22º in winter season.

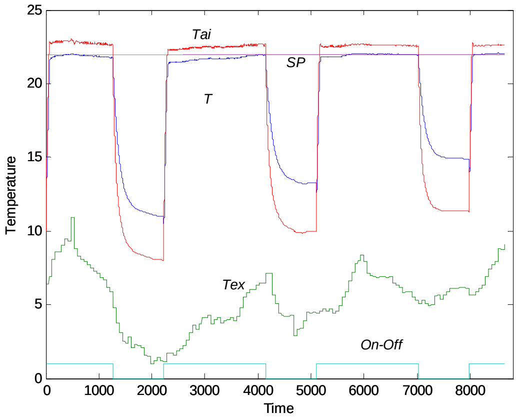


Figure 5: Simulation results with a 3-way-valve regulator

A model-based predictive control was also built. The aim of the zone model was to establish a strategy that would simultaneously reduce deviations from reference signals related to occupancy levels for different periods and the control effort needed to achieve this. This translates into the minimisation of a quadratic cost, taken as a reference index in which the weights, the prediction, and control horizons and restrictions can produce an enormous variety of strategies. It is then possible that these parameters could be subjected to multiple-criteria optimization, depending on the specific goals to be achieved by the thermal system. More precisely, it would be possible to consider an advanced control scheme; a Constrained Model Predictive Control based on the model already identified. This would suggest the optimization of a performance index based on minimization of the following function:

$$V(k) = \sum_{i=H_w}^{H_p} \left\| T(k+i:k) - r(k+i:k) \right\|^2_{Q(i)} + \\ + \sum_{i=0}^{H_u-1} \left\| \Delta u(k+i:k) \right\|^2_{R(i)} \tag{4}$$

In order to achieve a good tracking of the zone temperature *T* with respect to the various desired temperature set points r, the prediction horizon and the prediction control horizon, H_p and H_u, must be estimated. In order to obtain energy savings, this must occur during the operation time and must apply limits to the control signals.

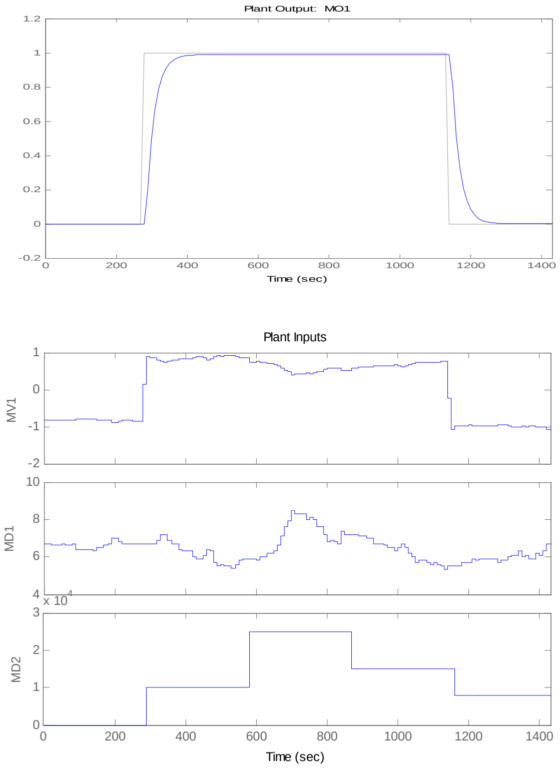


Figure 6: Restricted linear MPC room temperature

Two results are presented in this modern control design structure. The reference temperature profile is considered as step changes from a stationary state. The first is a restricted linear MPC with a linear zone model, where the control signal, manipulated variable, (*MV*) is the temperature of the air flow and the measured disturbances (*MD1* and *MD2*) are the external temperature and the zone occupancy (see Fig. 6).

The second is a restricted non-linear MIMO system: an MPC where the heat flow in the zone is not linear and where the control signals are the air-flow rate and the air-flow temperature (see Fig. 7). Simple interpretation of the results shown allows the thermal engineering to activate new control strategies with the perspective of comfort and energy saving.

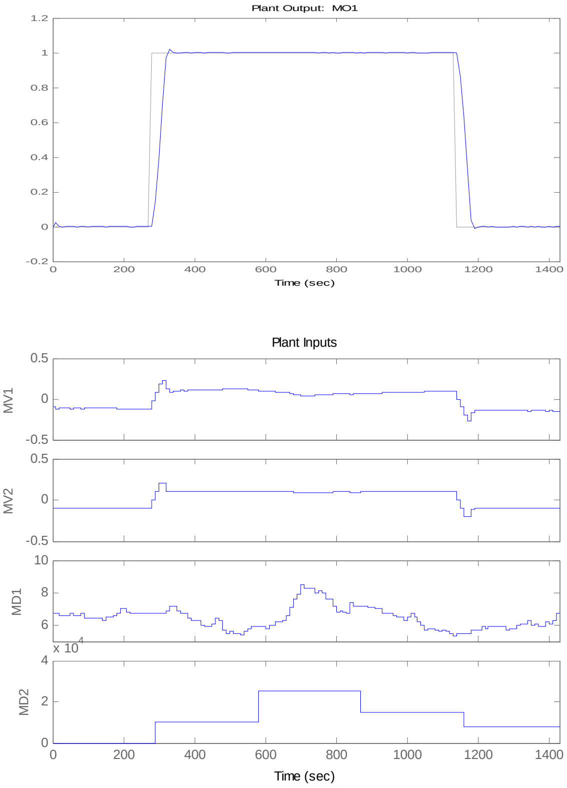


Figure 7: Restricted non-linear MPC room temperature

In order to achieve the best solutions, the parameters of the controller, such as the prediction horizon, the control prediction horizon and the weights in both the output error and in the control signals, *Q(i)* and *R(i)*, in the performance index must be selected in a multiobjetive index. Selecting the best control design parameters based on a new performance index, such as the L-infinite norm or ISE, has been done previously in activated sludge processes by some of the authors of this work (Francisco et al. 2005).

5. CONCLUSIONS

Here we have investigated a general procedure for deriving a dynamic model to control an HVAC system. The model consists of a zone, an air-handling unit, a production unit, and control elements. The mathematical models for the components are derived based on physical properties and element characteristics. Simulation of the complete HVAC system was analyzed both in real and modelled situations, with the observation of excellent coincidences. The good result obtained in this agreement is a hint as to how the operation of a HVAC system might be improved, even though some algorithms cannot be introduced into real control devices. Classic PID controllers and Model Predictive Controllers based on models were then analysed, which revealed an improvement in the transient behaviour and rejected the disturbances and hence contributed to comfort and energy saving.

REFERENCES

Andersen K., Madsen H., Hansen L., 2000. Modelling the heat dynamics of a building using stochastic differential equations. *Energy and Buildings 31*: 13-24.

Francisco, M., Vega, P., Pérez, O., 2005. Process Integrated Design within a Model Predictive Control framework. *Proceedings of 16^th IFAC World Congress*. July 3-8, Prague (Czech Republic).

García-Sanz, M., 1997. A reduced model of central heating systems as a realistic scenario for analyzing control strategies. *Appl. Math. Modelling 21*: 535-545.

In-Ho, Y., Myoung-Souk, Y., Kwang-Woo K., 2003. Application of artificial neural network to predict the optimal start time for heating system in building. *Energy Conversion and Managemeng 44*: 2791-2809.

Kuehn, T., Ramsey, J., Threlkeld, J., 1998. *Thermal Environmental Engineering*. Upper Saddle River, NJ: Prentice Hall.

Liao, Z., Swainson, M., Dexter, A., 2005. On the control of heating system in the UK. *Building and Environment 40*: 343-351.

Mathews, E., van Heerden, E., Arndt, D.C., 1999. A tool for integrated HVAC, building, energy, and control analysis. *Building and Environment 34*: 429-449.

Riederer, P., 2002. Room thermal modelling adapted to the test of HVAC control systems. *Building and Environment 37*: 777-790.

Stec, W. J., van Paassen, A. H. C., 2005. Symbiosis of the double skin façade with the HVAC system. *Energy and Buildings 37:* 461-469.

Tashtoush, B., Molhim, M., Al-Rousan, M., 2005. Dynamic model of an HVAC system for control analysis. *Energy 30*: 1729-1745.

MODELING AND EVALUATION OF ALTERNATIVE PRODUCTION SCENARIOS IN THE FIELD OF COMPOSITE MANUFACTURING

B. Scholz-Reiter[(a)]**, M. Lütjen**[(b)]

(a) University of Bremen, Hochschulring 20, Bremen, Germany, 28359, Bremen
(b) BIBA-IPS, Hochschulring 20, Bremen, Germany, 28359, Bremen,

(a)bsr@biba.uni-bremen.de, (b)ltj@biba.uni-bremen.de

ABSTRACT

At the perspective of production process engineering there is special requirements and characteristics for the manufacturing of composites like carbon-fiber-reinforced plastics (CFRP). This paper introduces a special framework for the conceptual production planning of series productions for CFRP-products. The submitted modeling concept has a semi-formal graphical notation, which provides the specification, analysis and evaluation of manufacturing of alternative manufacturing scenarios already in the planning phase. The option of an automatically generated simulation model enables the planning engineer to analyze the system performance of the modeled manufacturing scenario. This detailed analysis enables a further optimization of the scenario without additional expenditure. Apart from the documentation aspects of this modeling concept, the production process engineering will be improved by the management of alternative manufacturing scenarios.

Keywords: conceptual process planning, production process engineering, generic material flow simulation, CFRP, factory planning

1. INTRODUCTION

The highest potential of lightweight construction for aircraft structures or rather all kind of transport technology is provided by carbon-fiber-reinforced plastics (Froböse 2003). Even in relation to aluminum structures a weight reduction of up to 30% is possible, which leads to substantial economic and ecological advantages. Accordingly, the importance of carbon-fiber-reinforced plastics (CFRP) increased clearly in the past two decades for the sectors of aircraft construction, shipbuilding and automotive engineering. The market demands for shorter development times, further cost reductions and increasing manufacturing rates.

The further development of CFRP manufacturing technologies requests a set of new planning methods in order to fulfill the demand for high efficient production processes and systems. For economic reasons the use of carbon fiber composite structures is mainly focused on special applications at the high tech sector like the aerospace industry (Kleineberg, Herbeck and Schöppinger 2003). An important cost driver for the manufacturing process is the minor level of automation. Especially the handling of carbon fiber is difficult because of the low stiffness. Accordingly there exist only a few feasible handling solutions for fully automated production lines (Froböse 2003). An automated manufacturing process includes higher investment costs for machines, robots etc. than a conventional manufacturing solution. In conclusion the investment risk for automated manufacturing solutions is even higher. Therefore it is necessary to plan in a very accurate and proper way.

Also the development of new economical production technologies at the end of the 90's extended the spectrum of alternative manufacturing solutions for carbon-fiber-reinforced plastics substantially. Therefore it is necessary to identify a few good potential manufacturing solutions at early stages in order to concentrate the planning efforts on these. Against this background, a continuous evaluation and comparison of the alternative manufacturing solutions should be possible over the entire planning process. This requires a specific modeling and evaluation framework, which provides an appropriate specification of a manufacturing process and production system.

New product development concepts like simultaneous engineering need more communication and coordination between the interdisciplinary planning engineers. A visual modeling concept can support this kind of communication from the view of process designing significantly. The design of a manufacturing process which involves new materials and technologies requires synthesizing technical knowledge from a variety of sources and experts (Albastro et al. 1995). The early and creative phases of process designing demand for a quick and easy depiction in a semi-formal language to find a direct and common way of communication between all participants. In using a graphical notation for the model, it can be understood by all kind of experts and the production process is documented for future activities. Additionally the formal description is the precondition for a computerized analysis and evaluation. The developed modeling concept supports the very early phases of process designing. Alternative planning scenarios will be described, analyzed and evaluated in a structured and fast way.

2. PRODUCTION PROCESS ENGINEERING

The term of Production Process Engineering (PPE) is used in this paper for defining a special subject of Business Process Engineering (BPE) which is focused on production environment. The objective of process planning is to transform an idea into a saleable product. In dependence on the product and the overall production concept the planning activities differ (Halevi 2003). Presuming a series production of composite parts by building up new production lines there will be a longer phase of conceptual process planning. The high investment volume leads to higher planning efforts. The objective of conceptual process planning is to describe one or more possible manufacturing solutions and to determine the corresponding costs. Normally the manufacturing operations are not very detailed in this phase and only a rough cost estimation can be given. However it allows the engineers to get an idea of the process and its needed resources. A first and rough cost estimation allows the engineers to adapt their decisions regarding to the process costs. The best suitable solutions can be identified and for example progressed in a more detailed process planning. On the other hand the conceptual process planning allows the evaluation of manufacturability and the identification of cost drivers as early as possible in the design process.

2.1. Conceptual Process Planning Activities

The process planning is part of the product development process and determines how raw material is transformed into its desired form. Such planning activities can be aided by the use of concepts like e.g. digital factory or CAPP (computer aided process planning). In general these concepts are orientated to later phases of process design, which will address especially technical aspects. In distinction the conceptual process planning focuses more on the economic issues by selecting e.g. the best suitable production techniques (Patrick, Dantan and Siadat 2007). The conceptual process planning can be described by a sequence of activities, which have to be accomplished (Figure 1). The shown order can change, but should stick to the four defined phases:

- Creation of Process Chain
- Resource Assignment
- Creation of Procuction Structure
- Dimensioning of Production System

The first phase "Creation of Process Chain" is characterized by determining the production techniques, process chains and quality assurance methods in addiction to the product specification. Thereby for each activity a time module has to be created, which defines the process time in dependence to the product specification and for e.g. machine performance. By using process chain templates the modeling effort can be reduced for process sequences of typical CFRP-production techniques. The first step by designing a process chain is to create a sequence of manufacturing or assembly steps without assigning any resources. The idea for this procedure is the goal of getting new and innovative solutions by thinking without technical, economical or organizational restrictions caused of the resources (Zenner 2006).

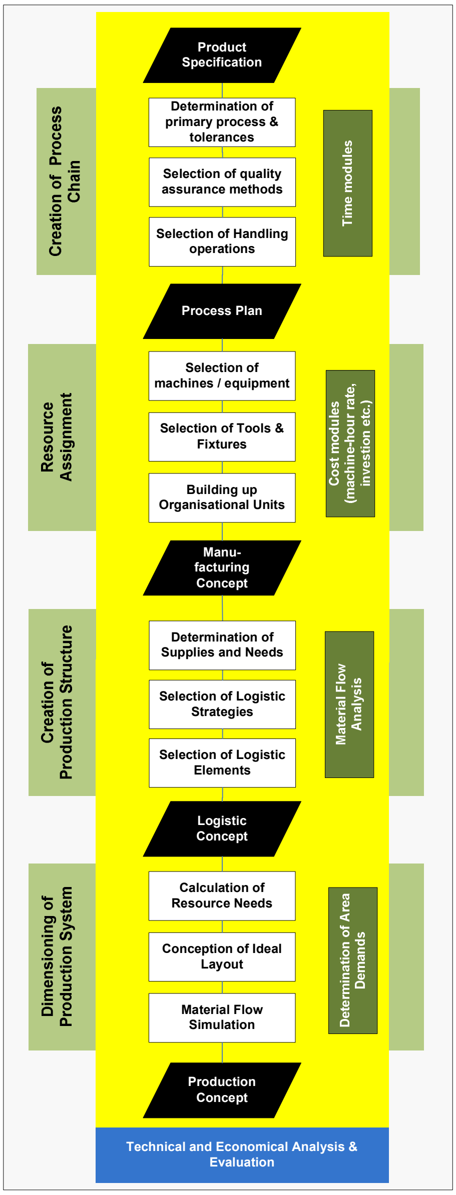

Product Specification
Determination of primary process & tolerances
Selection of quality assurance methods
Selection of Handling operations
Process Plan
Selection of machines / equipment
Selection of Tools & Fixtures
Building up Organisational Units
Manu-facturing Concept
Determination of Supplies and Needs
Selection of Logistic Strategies
Selection of Logistic Elements
Logistic Concept
Calculation of Resource Needs
Conception of Ideal Layout
Material Flow Simulation
Production Concept
Technical and Economical Analysis & Evaluation
Creation of Process Chain
Resource Assignment
Creation of Production Structure
Dimensioning of Production System
Time modules
Cost modules (machine-hour rate, investion etc.)
Material Flow Analysis
Determination of Area Demands

Figure 1: Conceptual Process Planning Activities

The second phase "Resource Assignment" is addressed to the selection of process compatible resources. This includes the machines and devices as well as the tool and fixtures. Especially the CFRP manufacturing needs an accurate design of handling operations, which normally includes the conception and construction of appropriate tooling, fixtures and handling devices. The assigning of worker is realized by creating organizational units which also include machines and equipment. By using or creating cost modules for each organizational unit, the allocation of direct and indirect costs is realized. The manufacturing concept contains a rough estimation for process time as well as for investment and operating costs.

The third phase "Creation of Production Structure" is characterized by designing the logistic concept. Therefore the manufacturing concept is projected onto the production structure. Thereby material flow connections between e.g. machines or organizational units emerge. According to the classic logistic functions of transshipment, storage and transport this production structure is extended with logistic elements. Based on material flow analyses, compatible logistic strategies are selected. With the use of logistic elements like e.g. buffers, storages and transport systems the logistic concept is modeled. This kind of modeling is only qualitative and describes only material flow connections. Additional data about the information flow can be noted but will not explicit be considered in the modeling concept.

The fourth phase "Dimensioning of Production System" is the ordinary field of factory planning. The objective is to determine the resource needs and to create a draft layout. The determination of resource needs can be done first by static calculation and then more detailed and accurate by dynamic material flow simulation. Because of stochastic deviations the use of a dynamic simulation model can help to make a good dimensioning of especially buffers, tools and fixtures. The identification of temporarily "bottle necks" can help to adjust and fine-tune the production structure in order to reach a better system performance. The result of this phase is a production concept which contains the numbers and capacities of all manufacturing and logistic objects as well as the alignment in a draft-layout. Because of having many handling operations with tools and fixtures, it is useful to layout the production concept in order to get accurate area demands. Normally the additional area demand for handling operations is difficult to estimate without a layout.

After defining the production concept, a technical and economical analysis and evaluation will be done. Normally a set of different planning scenarios will be tested and compared. Because of having strong technical restrictions, the automatic generation of alternative planning scenarios for e.g. different product designs is more or less impossible. Especially the estimated process times would change in dependence on the product design specification. Therefore an extensive optimization of alternative planning scenarios with specific operations research models is not done.

2.2. Management of alternative Planning Scenarios

In general a lot of suitable solutions will exist for a planning problem. The planner's objective is to find the best solution in an acceptable planning time. Depending on the planning problem, various possible solutions differ in the achievement of planning objectives like e.g. cost, quality, flexibility, time. Especially the design of large-scale production systems can multiply little differences to big amounts. Therefore a planning engineer should be assisted methodically by the management of alternative planning scenarios in order to find the best solution. Normally, generating of ideas and approaches is much easier than reducing and finding the best appropriate alternatives. Every alternative planning scenario which is further determined increases the planning expense (Bley and Zenner 2005). Therefore, an early abort of adverse planning scenarios is needed. For the special field of process designing the following level-concept of a Planning Pyramid for alternative Process Designs is proposed (Figure 2).

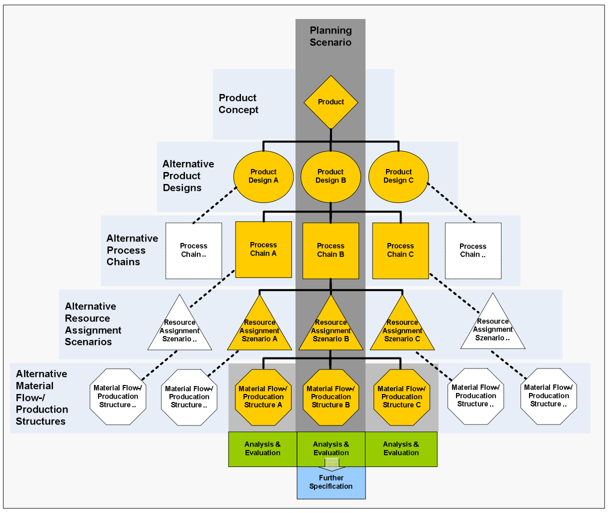

Planning Scenario
Product Concept
Product
Alternative Product Designs
Product Design A
Product Design B
Product Design C
Alternative Process Chains
Process Chain ..
Process Chain A
Process Chain B
Process Chain C
Process Chain ..
Alternative Resource Assignment Scenarios
Resource Assignment Szenario ..
Resource Assignment Szenario A
Resource Assignment Szenario B
Resource Assignment Szenario C
Resource Assignment Szenario ..
Alternative Material Flow-/ Production Structures
Material Flow-/ Producation Structure ..
Material Flow-/ Producation Structure ..
Material Flow-/ Producation Structure A
Material Flow-/ Producation Structure B
Material Flow-/ Producation Structure C
Material Flow-/ Producation Structure ..
Material Flow-/ Producation Structure ..
Analysis & Evaluation
Analysis & Evaluation
Analysis & Evaluation
Further Specification

Figure 2: Planning Pyramid

Every manufacturing process is designed for a defined type of product. Therefore the topmost level of the planning pyramid is used by the product concept. The second level contains alternative product designs. Normally these two levels are assigned to a product design or construction team. The following levels describe the actual field of process designing. The level of alternative process chains defines different sequences of process steps for manufacturing and assembly. Usually the alternative process chains differ in manufacturing technology and method. At the next level every process step is assigned to necessary process resources like e.g. machines, tools and workers. In order to reduce the possible assignments it is useful to define resource assignment scenarios which determine one special configuration of assigned resources. At the last level alternative material flow and production structures are defined which determine e.g. layout, transport and storage solutions.

The different levels of the planning pyramid represent various elements of planning. A planning scenario contains exactly one element of each level. Each of these planning elements can be specified more detailed on its own. The first evaluation of a planning level would base on rough guess and get more precisely during the planning process. The evaluation and decision about the best suitable alternatives would lead to further planning activities and specifications.

2.3. Cost Evaluation and Analysis

In order to calculate proper part costs an analytical approach of cost evaluation is selected. Such an analytical approach is the most accurate method for cost evaluation (Patrick 2007). In distinction to other approaches like e.g. analogical and parametric ones which orientate to aggregated parameter models, this approach defines cost modules for each organizational unit. Such a cost module defines variable and fixed costs for the organization units. In dependence on the workload the hourly rates can be calculated. For imitating more product types using the same production line, the workload and the process costs can be adjusted. Especially the fixed costs have to be allocated to the different product types. In dependence on the level of detail an organizational unit can consist of only one machine or a production cell with a lot of machines and workers. The basic cost types of an organizational unit are:

- labor costs
- resource costs
- supply costs
- area costs

By differentiating logistic and process costs, the factory and layout planning can be assisted (Figure 3). Especially the logistic costs are depending of the selected material flow and layout solutions. Therefore the factory planner can compare different factory solutions and e.g. improve the factory layout. The logistic costs consist of transshipment, transport and storage costs. The process costs are the indicator for the quality of the manufacturing concept. For making better analysis, the process costs can be further divided into manufacturing, assembly and inspection costs.

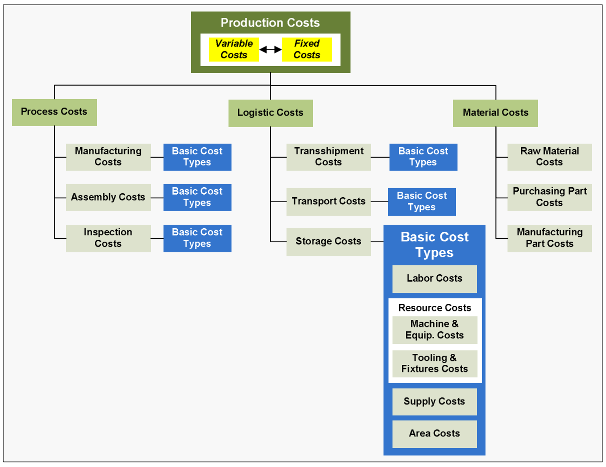


Figure 3: Ontology of Cost Modules

In order to calculate the production costs also the material costs have to be recognized. The material costs consist of raw material costs, purchasing part costs and manufacturing part costs. The production costs per part can be easily calculated by multiplying the hourly rates with the estimated process times. Because of having dynamic process times for the logistic elements, these costs cannot be calculated without identifying the actual using rates of different product types. In the case of having only a one-product production all costs can be converted directly into costs per part and different planning scenarios can be compared very easily.

3. COMPOSITION OF MATERIAL FLOW OBJECTS

For the development of the modeling concept it was necessary to identify the main logistic characteristics of CFRP manufacturing. One of these characteristics is the use of tools like e.g. molds for shaping the working pieces. In differentiation to other manufacturing technologies the molds follow the material flow and serve as carriers. Furthermore, auxiliary material like e.g. vacuum bags enter the manufacturing process, follow and leave the material flow after a while. Another important aspect of CFRP manufacturing is the ability of combining different parts in a manufacturing sequence. This is normally a characteristic for assembly processes. In conclusion it is necessary to define stages of production for each part in order to get a transparent depiction of auxiliary materials and tools in the material flow. This characteristic will be named in this paper as the composition of material flow objects.

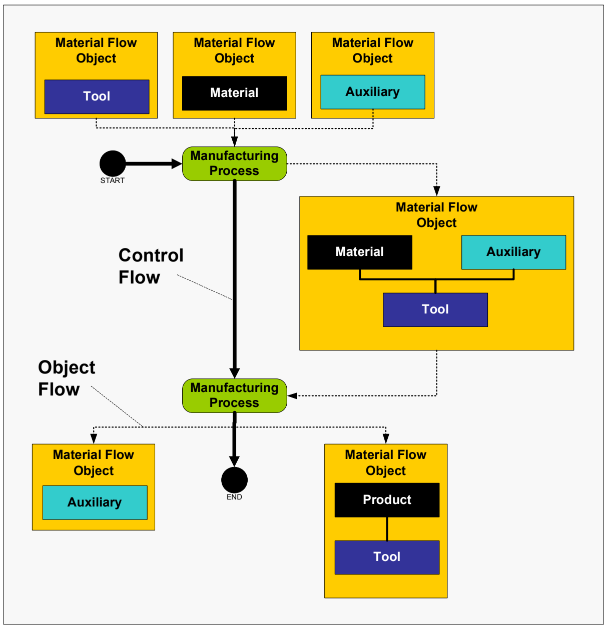


Figure 4: Modeling the Object and Control Flow

By defining the requirements for the modeling of CFRP manufacturing processes it was noticed that the common enterprise modeling techniques like e.g. CIMOSA, IEM, UML allow the modeling of material flows, but not in such a detailed way like needed (Bernus 2005). Also the most popular software solutions for digital factory like e.g. DELMIA, UGS are not able to fulfill the special requirements for CFRP manufacturing. Therefore a graphical modeling concept was developed which orientates to the common enterprise modeling techniques, but is extended to the ability of combining different material flow objects to new instances of material flow objects.

Every passive resource like e.g. tools, materials which enter a process step is defined as a material flow object (Figure 4). Several material flow objects can be combined to only one material flow object. The composition of material flow objects orientates to the physical part connection. In addition to the object flow a control flow is used for the planning of the process sequence. This control option is very basic but acceptable for defining process chains of series productions. The predecessor and successor of process steps are defined in this way.

4. GRAPHICAL MODELING CONCEPT WITH PROCESS AND STRUCTURE VIEW

The high complexity of production processes engineering requires a view concept for modeling. The view concept serves as an expedient to reduce the complexity of model designing (Scheer 1994). Therefore two basic views are provided, which differ in process and system view (Figure 5). The vision is to enable every engineer to read and understand the model with less introducing.

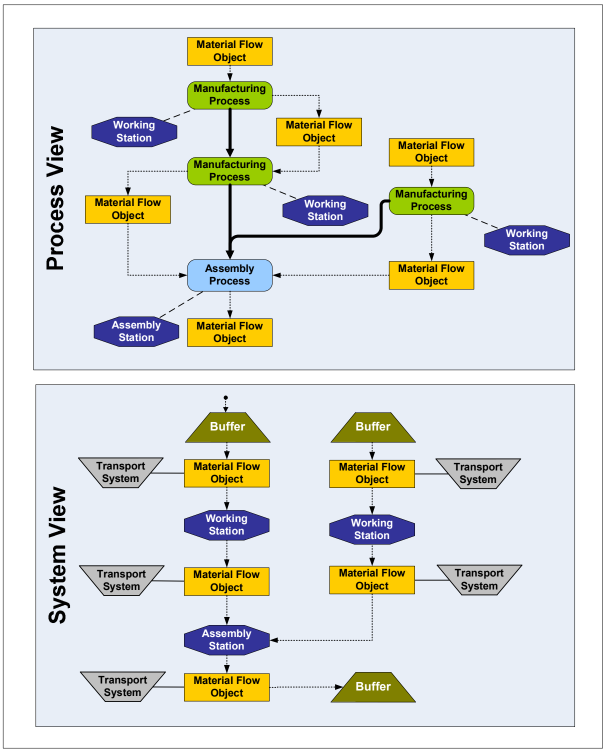


Figure 5: Process and Structure View

The process view is focused to the process structure. First the process sequence is modeled by connecting the process steps with each other. Then the material flow is added. By assigning the resource elements to the process steps, the manufacturing concept will be created. Every process step has to be assigned to a processor. The processors are the basic elements of the system view. They are connected by material flows. In general the processor is an organizational unit. The system view can be generated from the process view. The system view describes the structure of the production system and is mainly addressed to the logistic implementation of transport and storage systems. By adding logistic elements to the system view, the both views will get inconsistent. The process view contains no logistic processes in order to focus on manufacturing aspects. In spite of the both views are linked by the processors and material flow objects. Therefore any change in the process view will lead to changes in the system view. Various processes of different product variants can be integrated in into same system view. Every physical transport between two processors should be assigned to a transport system.

5. GENERIC MATERIAL FLOW SIMULATION MODELS FOR DIMENSIONING

Material flow simulation models help to understand the behavior and structure of logistic systems. They are the preferred planning tool for analyzing and designing such systems. But there is a problem with simulation models. The development of such a model needs a lot of time and is highly complex. Finally, a real or a planned system is described by the simulation model. Nevertheless, it is not enough to develop an accurate simulation model. The model must be understood, updated, reused and inhered by others. Normally, the simulation results are written down in separate documents but the simulation model on its own is not documented. Most of the knowledge about the model is directly linked to the simulation engineers and can be lost. Therefore it will be necessary to document the simulation model itself (Oscarsson and Moris 2002).

For this reason a standardized documentation would help to explain the model in order to describe the underlying processes, controls and logics. Especially the visual depiction of process sequences in a detailed way, how it is needed for the field of process designing, cannot be done in a simulation model directly. All relevant aspects for generating process and system structures can be modeled with using the process and structure view. Therefore, modeling is more transparent and can be better communicated to other participants. In differentiation especially to the business process modeling techniques, this concept focuses on material flow modeling and less on information flow modeling. Therefore only some basic control flows like e.g. process sequences are graphical modeled and the more complex control and decision procedures have to be programmed separately in the simulation software.

In distinction from efforts of creating a common simulation language like e.g. SRML (Simulation Reference Markup Language) the submitted approach orientates to the modeling of production structures and processes by using a graphical notation. For doing material flow simulation such a graphical model has to be converted into a simulation language/model.

The advantage of such a material flow oriented graphical model with only simple information flows is, that it could be easily compiled into different simulation tools/languages. Normally the integration of planning and control operations is more challenging, but the creation of the basic production structure is also very time consuming. The preferred discrete simulation tool for the implementation of this framework is DELMIA Quest. The inner material flow simulation structure of DELMIA Quest corresponds in an ideal way to the developed graphical modeling concept (Figure 6). By assigning process steps to the corresponding processors like e.g. workstations, the material flow simulation structure can be compiled very easily and fast.

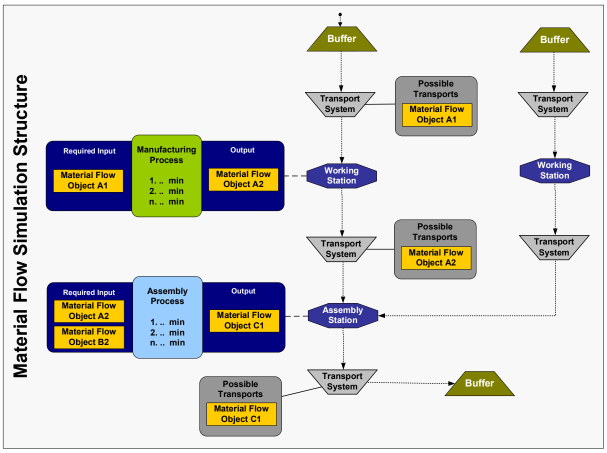


Figure 6: Material Flow Simulation Structure

Additionally to the graphical notation an object oriented database software solution has to be developed. Not all production process and system information can be added to a diagram. Also the different planning scenarios have to be managed. A first prototypical implementation is done by using yEd Graph Editor, MS Access and DELMIA Quest. The diagram is modeled in the yEd Graph Editor and transferred to MS Access via XML. The simulation model is created in MS Acecess by using the scripting language of DELMIA Quest. The project's implementation has shown, that especially the management of alternative planning scenarios by using resource and process libraries is extremely challenging.

6. CONCLUSIONS AND FUTURE WORK

Especially in aerospace industry the CFRP product development process is characterized by complex planning problems. The possibility of integral part manufacturing enables many alternative product and process designs. In order to find the most promising design and manufacturing solutions, various alternative planning scenarios have to be defined, specified, analyzed, evaluated and compared to each other. The management of so many scenarios is very difficult, because each scenario has several iteration loops. In result a lot of changes have to be accomplished and analyzed in the planning model. Therefore a framework for the management and evaluation of alternative planning scenarios is developed and presented by this paper.

Additionally, this paper provides an approach of a object oriented modeling concept with a semi-formal notation, which is focused on production processes and material flows. Such a model supports the analysis and evaluation of alternative production scenarios by the option of material flow simulation. A basic material flow simulation model can be generated for external discrete simulation tools. Regarding to future works the full implementation of the concept in a prototypical application is planned. Additionally, the view concept should be enhanced for the subjects of quality management. Especially the importance of quality management in aerospace industry leads to the need of an integrated designing approach for creating compatible manufacturing and inspection processes.

REFERENCES

Albastro, M., et al., 1995. The Use of Visual Modeling in Designing a Manufacturing Process for Advanced Composite Structures. *IEEE Transactions on Engineering Management* 42 (3): 233-242.

Bernus, P., et al., 2005. *Handbook on Architectures of Information Systems*, 2nd ed. Berlin/Heidelberg: Springer.

Bley, H., Zenner, C., 2005. Handling of Process and Resource Variants in Digital Factory. *CIRP Journal of Manufacturing Systems* 34:187-194.

Froböse, R., 2003. *Automatisierung erschließt Karbonfasern neue Potenziale*. Innovations-report. Available from: http://www.innovations-report.de [accessed 01 May 2007]

Halevi, G., 2003. *Process and Operation Planning: Revised Edition of the Principles of Process Planning, A Logical Approach*. Dordrecht: Springer.

Kleineberg, M., Herbeck, L. Schöppinger,C., 2003. Industrialisierung der Prozesskette Harzinfusion. *9. Nationales Symposium der Sampe*, February 2003, TU Clausthal (Clausthal, Germany).

Oscarsson, J., Moris, M., 2002. Documentation of discrete event simulation models for manufacturing system life cycle simulation. *Proceedings of the 2002 Winter Simulation Conference*, 1073-1078. December 2002, San Diego (California, USA).

Patrick, M., Dantan, J.-Y., Siadat, A., 2007. Cost Estimation and Conceptual Process Planning. In: Cunha, P., P. Maropoulos, eds. *Digital Enterprise Technology: Perspectives and Future Challenges*. New York:Springer, 243-250.

Scheer, A.-W., 1994: *Business Process Engineering Reference Models for Industrial Companies*. Berlin: Springer.

Zenner, C., 2006. *Durchgängiges Variantenmanagement in der Technischen Produktionsplanung*. Thesis (PhD). Universität des Saarlandes.

ENERGY EFFICIENCY IN A STEEL PLANT USING OPTIMIZATION-SIMULATION

Ivan Ferretti[(a)], Simone Zanoni[(b)], Lucio Zavanella[(c)]

[(a) (b) (c)]Mechanical and Industrial Engineering Department
Università degli Studi di Brescia
via Branze, 38 – I-25123 BRESCIA (Italy)

[(a)]ivan.ferretti@ing.unibs.it, [(b)] zanoni@ing.unibs.it, [(c)]zavanell@ing.unibs.it

ABSTRACT

The recent years showed a significant increase both in energy costs and social awareness for environmental concerns. In the industrial sector, these aspects affected revenues and sustainability of the most intensive energy plants, such as steel mills. The present paper describes how simulation may powerfully support production planning and the related decisions. The application proposed refers to a steel plant feeding its Continuous Casting line by an electric Arc Furnace. The optimization of the production plan is pursued by an objective function which takes into account both traditional targets (e.g., lead times and due dates) and the need for an efficient use of energy. To this end, a discrete/continuous simulation model was developed: stochastic laws, suitable for temperature variations modelling, were implemented to forecast energy requirements at the Ladle Furnaces and/or the Vacuum Degasser. Final results showed how the simulation-based Decision Support System may lead to a significant reduction (more than 20%) of the daily consumption of electric energy.

Keywords: Simulation, Optimization, Energy efficiency.

1. INTRODUCTION

In the most recent years, the need for a more rational and efficient use of energy has emerged as a strategic and urgent issue. Such a necessity is particularly perceived in the industrial sector, not only because of the increasing costs of energy, but also as a consequence of the global competition, which stresses some features of the process and its final products (e.g., cost and quality). Furthermore, the rational use of the energy resource may be regarded as a twofold issue, a first aspect being related to the achieved consciousness of the limited availability of energy, regarded as a source, and the second being represented by a mature appreciation of the costs born to procure energy. In fact, according to a wider and wiser view of the problem, the term "cost" is to be appreciated not only in economic terms, but also in its social and environmental features. Therefore, the concept of "sustainability" is to be accepted and introduced in the industrial practices, as a consequence of the need for a limitation in the amount of energy required by human activities, so as to avoid to compromise the future of next generations.

In a similar scenario, "energy saving" covers a fundamental role, as this approach to industrial processes focuses on the capacity to control activities and practices reducing energy wastes (Cheung and Hui, 2004). Such a concept introduces a peculiar dimension of energy efficiency, as it implies a reduced use of energy sources, possibly integrated by the valorisation of energy residuals, to carry out the same activities. The aim of the present research is to show the development and the implementation of a model suitable to increase the energy efficiency in industrial plants, with specific reference to the case of steel mills. Regardless of the industrial sector taken as reference, the approach presents a more general validity, as the final result of the research shows how the simulation of the industrial process may provide an effective support to the formulation of energy-efficient and sustainable production plans. In general, steel plants are subjected to significant costs, due to energy consumption, which, therefore, highly influence both product and service costs: increasing energy efficiency may be an ineluctable way to maintain competitiveness. The topic described is particularly relevant to the Province of Brescia (Northern Italy) where the steel production is carried out mainly by mini-steel plant with Electric Arc Furnaces (EAFs). Presently, the largest part of these industrial structures are organized according to "mini-steel" plant schemes, which allow production plans arranged on small batches of even differentiated steel types. The type and quality of the steel produced may be varied several times per day. Such an increased flexibility is necessary to match market demand, as a make-to-order approach is generally considered as the most effective production strategy, given the high costs of raw materials and semi-finished products which have magnified stocking costs. The energy-related components significantly contribute to the overall accounting of production costs and, in particular, the need for differentiated production, short lead times and precise delivery dates have further determined the industrial interest for the mini-steel plant configuration. However, a similar approach also implies the need for advanced tools, so as to support production planning and decision making in a complex environment characterized by several parameters, numerous interacting variables and mutual constraints.

In the present paper, the tool adopted to support the process optimization is simulation. In fact, according to the Authors' experience, continuous simulation may provide a successful approach both at the design stage and in the running of existing plants (Gorlani and Zavanella, 1993). In particular, simulation allows the definition and setting of the large amount of variables, linked by complex relationships, which describe in detail the industrial process.

2. BACKGROUND

The literature in the field of industrial plant energy efficiency may be essentially divided into three categories:

1. Models for the plant design optimization
2. Models for the operational plant management: scheduling, maintenance, etc..
3. Models for the optimised plant revamping

It is not possible at all to find models universally applicable, they are mainly devoted to specific sector and often designed "ad hoc" for a given case study. The differences between the various types of industry requires a distinction to be effective and potentially applicable. The area most affected by studies is the petrol-chemical: this outcomes is easily justified. One of the most recent issue is related to the effort on how to extend the current analysis to all areas now that the energy problems are increasingly important in each sector.

The most frequently used technique for the optimised design of total site utility systems is the tool proposed by Papoulias and Grossmann (1983), that is mainly based on a super-structure of the system.

The models for the operational management are mainly focused on the problems of choice in the distribution of the load on units of service and the scheduling of maintenance. The problem is mainly treated recurring to mixed linear programming models (MILP). Examples refers to the work of Iyer and Grosmann (1997, 1998): the objective is the minimization of costs on an annual basis, through the search for a multi operational programming optimum period.

Another approach, in the literature, is the simulation. In the case illustrated by Prasad Saraph (2001) has tried to solve the problem of water in a biotech industry: the variability of industrial process together with a lack of planning for the provision of service led to interruptions in production due to lack of water.

One of the most popular techniques for the plant revamping optimisation is the pinch analysis. The primary objective of pinch analysis, regardless of the field where it is applied, is to optimise the use of materials or energy, achieving economic and environmental benefits. A powerful tool for optimizing energy in the industrial, which also exploits the pinch analysis, is the SitEModellingTM (Wolff et al., 1998)

In the literature on steel production there are several paper focused on energy reduction of the plant, on of the most relevant is Larsson and Dahl (2003) where a model based on an optimising routine is proposed. The main idea is to make a total analysis method for the steel plant system including the surroundings. The model is used to analyse the different possibilities for energy savings and practice changes within the system. The effect of optimising the total system versus separate optimisation of the different sub-processes is illustrated.

3. PROCESS DESCRIPTION

In this chapter the general process of a mini-steel plant will be described, according to the flow diagram reported in figure 1.

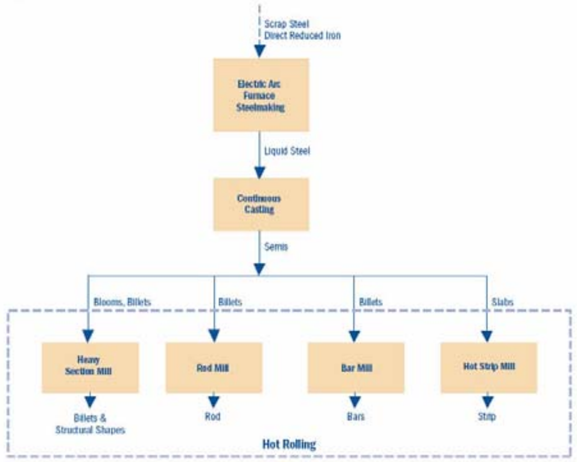


Figure 1: EAF Plant Production – Flow Diagram

Currently the percentage of the total electrical steel mills of European production is 35.3% (Stahl, 1997). In Italy and Spain production with electric arc furnace is considerably greater than that from blast furnace and converter oxygen. The main raw material in mini-steel plant with Electric Arc Furnace (EAF) is mild steel scrap which is procured from local and international markets. For the process description it is important to make a distinction between the production of ordinary steels for medium-low carbon alloy and production of special steel alloys (such as stainless steels). In the EU approximately 85% of the product is the first type. Thus we focus our attention on the description of special steel production where additionally operations are needed. The scrap is mixed in pre-determined proportions in the scrap yard and fed to the furnaces in charging buckets and melted by Electric Arc using Graphite Electrodes. The molten metal is processed to remove the impurities like sulphur and phosphorous and is subjected to slag off and further refining by adding Ferro alloys and other fluxes to bring it to the required standard specifications.

The molten steel is tapped at the required temperature to the pre-heated ladles. Steel ladles are equipped with latest slide gate opening system. Temperature of the molten metal in the ladle is measured to ensure correct temperature at the continuous casting machine. The liquid metal is then poured from ladle to the tundish and then to the water cooled copper mould on continuous casting machine. There takes place the billet formation by solidification of the molten steel due to water cooling. Billets coming out of the continuous casting machine are cut to the required length by gas cutting. The Billets are further rolled and converted into constructional steel of various section at Rolling Mills

Analyzing the steel flow diagram of the mini-steel plant oftenly the bottleneck of the process and thus the phase that dictate the cadence to the whole production program is represented by th continuous casting step.

3.1. Energy Needs

A typical energy balance for a modern EAF is shown inFigure 2.

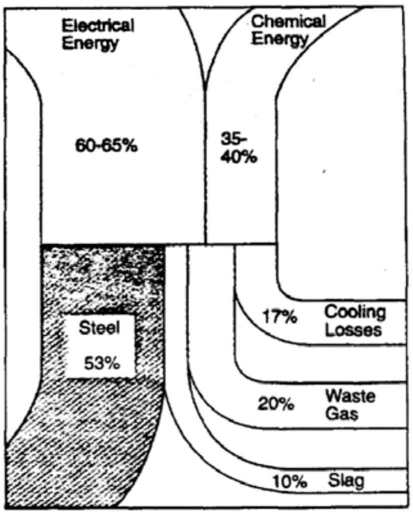


Figure 2: Energy Patterns in an Electric Arc Furnace.

Depending upon the melt shop operation, about 60 to 65% of the total energy is electric. About 53% of the total energy leaves the furnace in the liquid steel, while the remainder is lost to the slag, waste gas, and cooling. Just a decade ago tap-to-tap times (the difference time between the arrival time and the departure time of a ladle in the continuous casting station) had decreased from over 2 hours to 70-80 minutes for the efficient melt shops. Continuing advancements in EAF technology now make it possible to melt a heat of steel in less than one hour with electric energy consumption in the range of 360 to 400 kWh/ton. EAF operations utilizing scrap preheating such as the CONSTEEL Process can achieve even lower cycle Times and consequently a lower Energy consumption.

4. THE SYSTEM

4.1. Company description

The company considered as reference for this work (not explicitly mentioned for privacy) operates in the steel sector since 1933 in Northern Italy. At the begin it produced reinforced-concrete rod, afterwards it entered in the wire rod production field. In the following years it concentrated its effort on the production of special and high quality steels developing a product range for specialized applications. Its controlled casting using electro-magnetic stirrers enables it to offer products with very restricted analytical dispersions, a low concentration of sulphur, phosphorous and non-metallic inclusion contents and a low level of central segregation. The correct use of the Stelmor cooling system, ensures that its plant achieves precise mechanical properties and the degree of drawability required. Thanks to this control, it is able to obtain scale suitable for the pickling process required by the client. The plant is composed of the rolling mill department and the furnaces department, separated by a warehouse. The first department is composed of:

- In-line continuous steel rolling with horizontal and vertical stands and finishing cylinder housing in order to obtain section regularity and reduced dimensional tolerance.
- Controlled cooling in order to obtain homogeneous mechanical characteristics and consistent metallographic structure.
- Hot rolled steel bars with the option of thermomechanical rolling.

The furnaces department has the sequent plants:

- Electrical Arc Furnace (in brief EAF) (70 ton) that permits maximum control of chemical composition, minimal impurity content and homogeneity analysis.
- Two Ladle furnaces that permit quality improvement and a low inclusion level.
- Continuous casting plant with five strands that permits major ductility for resistance and total structure homogeneity.
- Controlled and ground with grinding machines to eliminate all surface defects.

We take into account only the optimization of the billets production in the furnaces department. The set of the products counts more than 400 different references. The sequent list presents the main references categories.

- Carbon steel for drawing applications ropes, pre-stressed concrete, spring: suitable for several industrial applications giving excellent performance in drawing and with a suitable surface for successive galvanizing treatments.
- Steels for welding applications: this grade of steel has the characteristic of high draw ability to facilitate fine wire drawing for electrode and welding applications with either gas or powder protection.
- Steels for case hardening, chains: these steels have high wear resistance and a relevant level of tenacity making them suitable for application in the automotive industry for the production of bushes, hubs, pins and timing system components. Highly appreciated for the production of high resistance chains with thermal treatment.

- Steels for hardening and tempering surface hardening: these grades of construction steel are applied in every industrial sector, mainly in the mechanical industry for the production of components requiring a high resistance to stress-fatigue.
- Steels for springs: suitable for the production of coil springs and torsion bars used in the automobile and railway industries; these grades of steel must have a high yielding point and tenacity after quenching and tempering to withstand stress under flexi-torsion.
- Steels for fasteners and for hot and cold heading applications: suitable for hot and cold deforming for the production of screws, bolts, nuts and all specific components requiring mechanical working. Severe controls during steel - processing permits us to obtain products which are free from non-metallic inclusions and surface defects.
- Sulphur free cutting steels resulpherised steel: these steel grades give improved machinability improving tool life.
- Creep resistant steel and steels for low temperature applications: these steels are employed in the mechanical and petrochemical fields, principally in power stations; they are used for equipment that must function at a temperature of over 350° C.

Following we introduce the problem and the main constraints handled underlying the objectives of the project.

4.2. Problem statement

The problem taken into account consists in the improvement of the furnaces department performances. In particular the main objective is to reduce the electric energy consumption of the department respecting the MRP delivery dates. Given the complexity of the system analyzed, the solution proposed consists in the definition of a Decision Support System (in brief, DSS) that permits the simulation and the consequent optimal setting of the production variables without modifying the plant layout. As explained below, the production variables considered are particular settings that the workers can do on the machines in the furnaces department. The actual manual setting causes two main inefficiencies: first every worker sets locally the machine without considering the interactions among his choice and those of the other workers that operate on other machines; second different workers choose different settings at similar operative conditions obtaining different performances. The main task of the DSS is, from one site, to obtain a solution considering globally the interactions among the machines of the department and, from the other site, to reduce the performances variations changing the workers.

In the following section we describe the firm's production process and the main variables and parameters taken into account in the simulation project.

4.3. Company's processes

In this section we describe the machines of the furnaces department and their interactions rules following the production flow. The department under consideration produces different kinds of steels obtained using specific chemical-physical processes. The department is formed by an Electrical Arc Furnace (EAF), two Ladle Furnace (LF) and a Continuous Casting plant with five strands. For the production of special steels it is used a Vacuum Degasser (VD). The typical production process considers the metal scraps fusion using the EAF and the following casting in the ladle. Then the ladle is transported by a overhead crane to the LFs where the casting is refined. At the end the ladle is transported to the CC where the different kinds of billets are produced. For particular kind of steels, it is necessary the vacuum degassing. This process considers, after the refining in the LFs and before the CC process, the transport to the VD. Every kind of steel has a specific set of times and temperatures for every station presented in the production flow. The model begins with the load of the metal scraps in the EAF and ends with the cutting of the billets in the CC station.

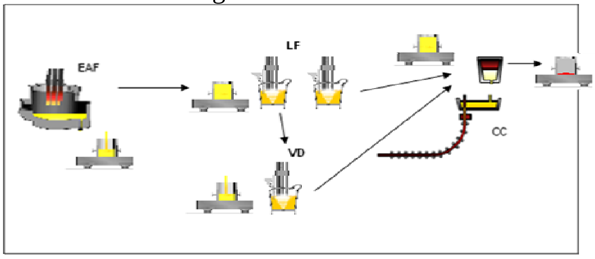


Figure 3: The furnaces department

Moreover, in order to understand correctly the system analyzed, it is important to consider that the CC station is the bottleneck of the department, because it has a production capacity lower than the other stations. In order to produce correctly, the castings have to arrive at the different stations with specific temperature values. Considering the loss of heat during the transport among the stations, this condition implies the correct schedule of the production times at the different stations. It is important to consider that only in the LFs the ladles can wait for the successive operations introducing an extra time, named holding time, after the refining process. In particular every LF can have at the same time only one ladle. Another important point to take into account is the fact that the EAF has to be blown out the minimum possible number of times. In fact the EAF firing uses many electric energy and the furnace requires many time to attain the right temperature. Finally the department has only one overhead crane for the ladles transport. Summarizing the main constraints considered are:

- The plant layout can't modified.
- The electric energy used can't go aver a maximum peak.
- The billets production sequence is defined by the MRP system and can't modified.

- The production times and setup times at the different stations are given.
- The production arrival temperatures at the different stations are given.
- The production temperatures for specific operations (refining, casting, …) are given.
- It is possible to use only one overhead crane for the ladle transport.
- Other specific machines constraints as, for example, capacity of the EAF, minimum and maximum temperatures, maximum casting weight at the different stations and so on.

The variables that can be modified in order to obtain the minimum electric consumption are the different level of speed to the LFs, defined as the time used to reach a specific temperature. In particular, greater is the speed, lower is the time to reach a specific temperature and greater will be the electric consumption. Another important variable is the holding time that can be set at the LFs stations in order to wait for the availability of the CC station.

Summarizing, the objective of the DSS proposed is to set the speed and the holding time at the LFs in order to produce the sequence defined by the MRP system and reduce the electric consumption. In the following section we introduce the parameters modelizations

4.4. Definition of parameters modelizations

After the definition of the production process and in particular of the elements that define the furnaces department, it is necessary to explain the kind of modelization used to describe the chemical-physical processes at the different stations and during the ladles transport. In particular, the chemical-physical processes are assumed as statistical relations among the variables temperature and time. Given the difficulty to make specific chemical-physical rules that correlate these three kind of variables, respect to, for example, the kind of steel, we have defined statistical models derived from a large samplings respect to the variables of interest. The first step in the system modelization is the definition of the casting that is specified by the follow parameters:

- steel code, that specified the kind of steel casted;
- casting number;
- sequence casting number, that specified the number of the casting in the MRP sequence;
- casting weight, that is a specific ladle constraint;
- Vacuum Degasser code, that indicate in the production process the use of the VD.

Every casting is transported by a ladle in the system, so in the following we use the term ladle to specify the particular casting transported.

For the production process modelization we begin from the last process that is CC. Because the CC can't stopped, it is necessary that when a casting of the sequence is terminated, another ladle has to be at the entrance of the CC. In order to respect this constraint, we have defined a variable named tcc(t) that is the time (minute) before the end of the casting under process where the variables t is the clock time. Because it knows the process time pcc (minute), considering the time scc (minute) as the time that the ladle use from the EAF (first station of the system) to the entrance of the CC, is possible evaluate tcc(t) as expressed by the following formula:

$$t_{cc}(t) = s_{cc} + p_{cc} - t$$

The temperature of the casting during the process Tcc is imposed by the quality assurance. The second process that we analyzed is the refining process in the LF's. If we consider only one LF the time that the ladle waits before the CC process is defined as:

$$t_{LF}(t) = t_{cc}(t) - tr_{LF-cc}$$

where trLF-cc is the time used for the transport of the ladle from the LF to the CC. So the ladle remains at the LF until tLF(t) is equal to zero. The hLF time, named holding time, is the difference between the available time and the process time and it can be defined as:

$$h_{LF} = t_{LF}(t_0) - p_{LF}$$

where tLF(t0) is the time when the ladle enters in the LF and pLF is the LF process time. In particular the LF process time depends on the weight variations of the casting due to the adding of additives during the refining process. This time has to be always greater than 0. In the real system we have to consider two LFs. In this case the tLF(t) time and aLF time change. In particular, the expression for every LFs assumes this form:

$$t_{LF(NumLF)}(t) = t_{cc}(t) + C(i-1) - tr_{LF(NumLF)-cc}$$

where C(i-1) is the forecast time of the previous casting to release the CC. if we suppose, for example, that ladle k is in LF 1 and that ladle k+1 is in LF 2. The time to cast completely ladle k is C(k). So the tLF(2)(t) is defined as:

$$t_{LF(2)}(t) = t_{cc}(t) + C(k) - tr_{LF(2)-cc}$$

The temperature at the end of the refining process is defined as the difference between the Tcc temperature and the cooling during the transport to the CC. It can be expressed as:

$$T_{LF} = T_{cc} - tr_{LF-cc} \cdot \alpha$$

where is the cooling rate during the transport. The value of cooling rate, expressed in K/min, is defined by experiment executed in the plant analyzed. The choice of a linear rule to describe the cooling during the transport derives from the analysis of this real phenomenon. The last station evaluated is the EAF. The EAF can release the ladle after the melting process only if the LF is free. Otherwise the ladle processed could have a low temperature for the downstream stations. In the case of one LF the time available for the fusion tEAF(t) is defined as:

$$t_{EAF}(t) = t_{LF}(t) - tr_{EAF-LF}$$

where trEAF-LF is the time used for the transport of the ladle from the EAF to the LF. In the model presented, the fusion time is stochastic and the fusion starts only if tEAF(t) is lower than the fusion time evaluated for the cast considered. Otherwise it is necessary to wait for the new fusion. In case of two LFs the evaluation of the tEAF(t) is the same, but the tLF(t) is evaluated considering every time the LF with the ladle ready for the CC. In this case the tEAF(t) can be expressed as:

$$t_{EAF}(t) = t_{cc}(t) + C(i-2) + C(i-3) - tr_{LF-cc} - tr_{EAF-LF}$$

When the fusion in the EAF stars, one ladle is in the CC, one is in LF1 and one is in LF2. The fusion temperature TEAF changes for every casts and it is distributed stochastically. In the case of the use of the VD we have to redefine the tLF(t) and TLF considering this special process. In particular the new expressions became:

$$t_{LF(NumLF)}(t) = t_{cc}(t) + C(i-1) - (t_{VD} + t_{AV}) - tr_{LF(NumLF)\ VD} - tr_{VD\ cc}$$

$$T_{LF} = T_{cc} \quad (tr_{LF-VD} + tr_{VD-cc}) \cdot \alpha \quad t_{VD} \cdot \beta \quad t_{VD} \cdot \delta$$

where tVD and tAV are respectively the VD process time and other refining time after the degassing process and where and are respectively the cooling rate during the VD process and the other refining after the degassing. In the figure 4 we report a summarizing schema for the LF available time.

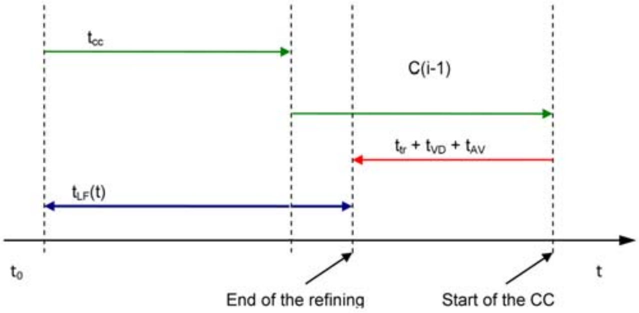


Figure 4: LF available time

In figure 5 it is represented how we have modelled the heating rate used to reach the target temperature and the cooling rate. As in the case of the cooling rate, we have modelled the heating rate with a linear rule validated by the collected data. Higher values of the heating rate permit to reach quickly high temperature, from the other site grater is the heating rate grater will be the electric consumption. Respect to the cooling rate that is fixed for the different kinds of steel during the simulation, the heating rate can be set manually by the workers changing the speed of the LFs. In this case it is not considered more than one heating, because of plant constraints.

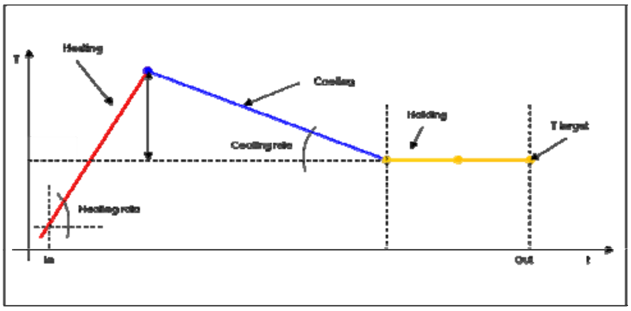


Figure 5: Temperature behavior in the LFs

The last parameter that we have to consider is the electric consumption. Given the complexity of the system analyzed, we consider only the electric energy lost during the holding time at the LFs and the number of EAF power offs. In the first case, it possible to identify a correlation between the holding time to the LFs and the energy consumption. More time the ladles are in the LFs more energy is necessary to maintain the temperature to the target temperature for the CC. In particular, considering a cooling rate in the LF equal to 0.5 K/min we can estimate the energy consumption as follow:

$$\Delta E = t \cdot \pi \cdot 32.70$$

where 32.7 is an experimental parameter that takes into account the heat dispersion of the LFs considered. Other energy saving can be obtained reducing the number of power off of the EAF. So, as in the previous case, instead of modelling the complex chemical-physical rules that correlate the temperature and the electric energy consumption for the entire fusion process, we have modelled only the number of power offs and considered only the saving derived from them reduction. Experimentally, we can consider the use of 700 KWh/min for every power off.

The process and transport times, the cooling rates and the temperatures described above are data of the model extrapolated from the real system. In particular we have executed a large sampling (extended for 6 months) that has permitted to define statistical models for the parameters under consideration. The process time at the EAF, for example, is a Weibull distribution with mean 51.12 min and standard deviation 4.22 min. In order to obtain these distributions we have used a input analyzer software (Input Analyser 8.01 by Rockwell Software®) that define the better distribution respect the data supplied. Another important parameter in the simulation is the casting weight. The casting weight depends on the substances added during the refining process and the dross. We have defined the distribution of this value analyzing more than 100 casts. The casting weight is set as a Normal distribution with mean 74,8 ton and standard deviation 2,87 ton.

4.5. Model validation

We have performed a validation for each station: CC, LF's, VD and EAF. Following we report the figures where we compare the real tap-to-tap times and simulated tap-to-tap times in the CC for a given representative casts set, used as reference for the validation. The figures show that the simulation is close to the real situation. In particular we calculate a mean error respect to the real data of 5.22%. The maximum distance between the simulated data and the real data is 7 min that represents the 15% of the real tap-to-tap CC time.

In the case of LFs we have compared the total simulated time with the total real time spent by the ladle in the LF station. This time takes into account the LF process time and the time spent waiting for the available of the CC. Following we report the graph related to the LF validation.

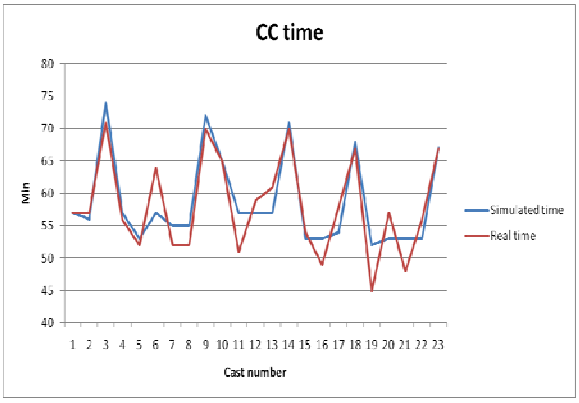


Figure 6: Validation of the CC time

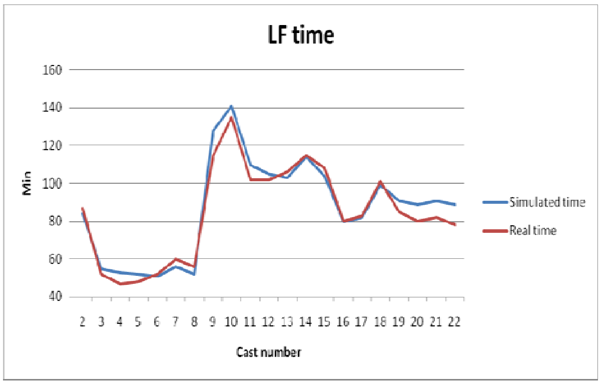


Figure 7: Validation of the LFs time

The mean error is 5.75%. Also in this case the simulated behaviour fits quite good to the real behaviour. In the case of VD we have evaluated the total simulated time with the total real time spent by the ladle in the VD station. In this case the mean error is 4.89%. Following we report the graph related to the model of the electric consumption to the LFs. In this case we have a mean error of 4,94%.

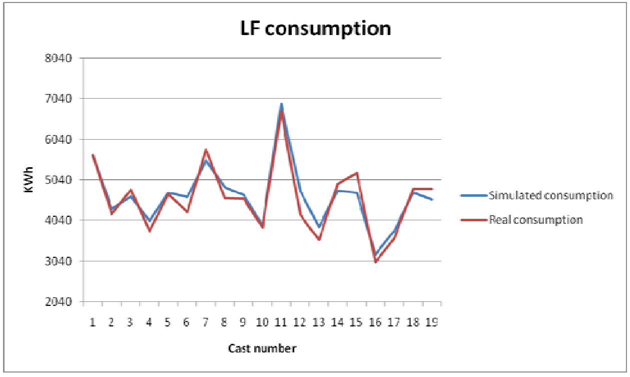


Figure 8: Validation of the LFs consumption

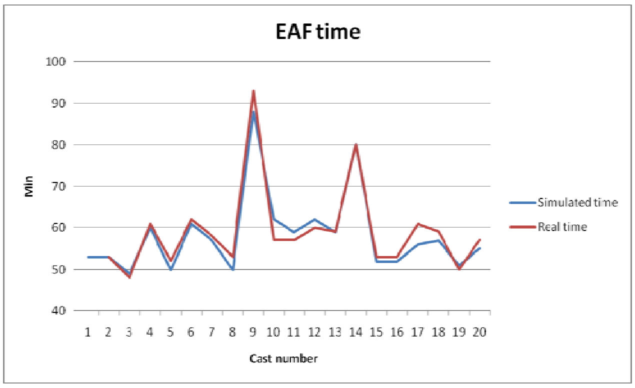


Figure 9: Validation of the EAF time

The last station validated is the EAF. In this case we consider the difference between the simulated fusion time and the real fusion time. In this case we have a mean error of 2.92%.

In the following section we introduce the optimization problem analyzed and the method used to solve it.

4.6. Optimization & tool

As already mentioned the bottleneck of the system considered is the CC: the CC process imposes the production rate to the other stations of the system defining the time windows where the ladles can start the process itself. Given this constraint it is important to optimize the total production lead time in order to reduce the waiting time to the LFs and the EAF power offs. In fact, we have to consider that when the ladles wait in the LFs or the EAF is blown out, the system consumes electric energy as reported above. In general, the inefficiencies are due to the fact that EAF, LFs and CC are not globally optimized. The variables that have to be optimized are:

- the LF speed: it is possible to choose seven different speeds (seven different heating rates);
- the holding time to the LFs: the minimum value of this time is 5 minutes.

In order to define the DSS we have developed a simulative model that represents the steel plant described above. The simulation model is coded in Arena® 8.01 (by Rockwell Software Inc.) simulation software. Based on the previous rules and associated parameters, the model simulates the processes to the different stations analyzed. In particular we have used a discrete/continuous simulation approach due to the necessity to simulate the heating and the cooling processes at the different stations. In order to obtain the minimum electric consumption given a specific production sequence, we need to determine the optimal control parameters defined above. For this purpose we adopt the optimization module OptQuest® 8.0 (Kelton et al., 2004) in Arena®. Each scenario is simulated for 1 production week (168 periods) and replicated 5 times. The number of replications shows to be quite enough to get a small variance in the simulation results.

4.7. Results

In this section we report the results obtained via the optimization of the simulative model described. In particular we show in figure 10 the electric energy consumption reduction obtained at the LFs.

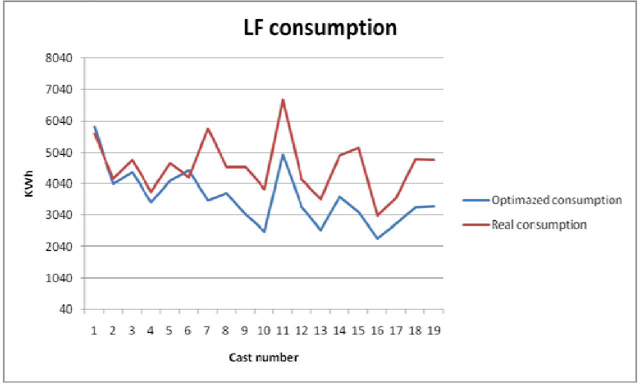


Figure 10: Comparison between the optimized electric consumption and the real electric consumption

It is important to note that with the first 6 casts the reduction of the electric consumption are not so significant because these casts have the VD process. In these cases the holding time is already reduced because the ladle has to go the VD for other operations. The only benefit of the optimization in the case of VD is the right choice of the LFs speed. Observing the results, we obtain a mean reduction of electric consumption of 21% at the LFs (18609 KWh). The adoption of this optimal strategy minimize the number of EAF power offs reducing the total time where the EAF is blown out (-21 min). This entails a reduction of the electric consumption of 14,700 KWh.

5. CONCLUSIONS

In the present work we have proposed a DSS that has the objective to set the speed and the holding time at the LFs in order to produce the sequence defined by the MRP system so as to reduce the electric energy consumption. Results obtained support company’s manager to optimize the usage of strategic leverage, since the DSS defines the solution taking into account the entire production process. In particular the overall electric energy saving is larger than 20%.

REFERENCES

Cheung K.Y., Hui C.W., 2004. Total site scheduling for better energy utilization, *Journal of Cleaner Production*, 12 (2), 171-184

Gorlani C. and Zavanella L., “Continuous simulation and industrial processes: electrode consumption in arc furnaces”, *International Journal of Production Research*, 1993, 31(8), 1873-1890

Iyer R.R., Grossmann I.E., 1997.Optimal multiperiod operational planning for utility systems, *Computers and chemical engineering*, 21, 787–800

Iyer R.R., Grossmann I.E., 1998. Synthesis and operational planning of utility systems for multiperiod operation, *Computers and chemical engineering*, 22, 979-993

Kelton W.D., Sadowski R.P., Sturrock D.T., 2004. Simulation with Arena, McGraw Hill.

Larsson M. and Dahl J., 2003. Reduction of the Specific Energy Use in an Integrated Steel Plant—The Effect of an Optimisation Model, *ISIJ International*, 43 (10), 1664–1673

Papoulias, S. A., and Grossman, I. E., 1983. A structural optimization approach in process synthesis-I. Utility systems, *Computers and chemical engineering*, 7(6), 695-706.

Prasad V. Saraph, 2001. Simulating biotech manufacturing operations: issues and complexities. *Proceedings of the 2001 Winter Simulation Conference*, Arlington- VA (USA), 530-524.

Wolff A., Groebel M.J., Janowsky R., 1998. SitEModellingTM: a powerful tool for total site energy optimization, *Computers and chemical engineering*, 22, 1073-1084.

AUTHORS BIOGRAPHY

IVAN FERRETTI, graduated in 2003 in Industrial Engineering at the University of Brescia (Italy). He took his Doctorate at the University of Brescia (Italy) in 2008. His main research interests are Simulation, Applied Meta-heuristics and Reverse Logistics. His e-mail address is : ivan.ferretti@ing.unibs.it.

SIMONE ZANONI, is currently an assistant professor in the Mechanical & Industrial Engineering Department at the University of Brescia (Italy). He graduated (with distinction) in 2001 in Mechanical Engineering and took his Doctorate at the University of Brescia (Italy) in 2005. His main research interests are: Layout Design, Inventory Management and Closed-loop Supply Chain. His e-mail address is : zanoni@ing.unibs.it.

LUCIO ZAVANELLA, is currently full professor in the Mechanical & Industrial Engineering Department at the University of Brescia (Italy). He graduated in 1982 in Mechanical Engineering at the Politecnico di Milano. Since 1986, he has been a researcher in the field of production systems at the Università degli Studi di Brescia. His research activities mainly concern manufacturing systems and inventory management, also with reference to environmental aspects. His e-mail address is: lucio.zavanella@ing.unibs.it.

OBJECT ORIENTED LIBRARY FOR DYNAMIC MODELING AND SIMULATION OF SUGAR HOUSES FOR OPERATORS TRAINING.

Rogelio Mazaeda[(a)], **César de Prada**[(b)].

[(a)(b)]System Engineering and Automation Department, Sciences Faculty, Valladolid University.
c/ Real de Burgos s/n, CP. 47011. Valladolid, Spain.

[(a)] rogelio@cta.uva.es, [(b)] prada@autom.uva.es

ABSTRACT

A reusable object oriented dynamic model library for the simulation of the sugar crystallization process is described. The Sugar House, which is the final department of a beet sugar factory, receives a stream of concentrated sucrose syrup and delivers the commercial product in the form sugar crystals. The intended purpose of the library is the creation of a simulation tool that contributes to the training of control room operators.

Keywords: process modelling, operators training, sugar

1. INTRODUCTION.

A typical beet sugar factory has the general structure shown in figure (1). The sucrose juice is extracted from the previously sliced beets in the first section by means of a diffusion process. The obtained green juice contains most of the sucrose originally brought in with the beets but it inevitably is contaminated with many other organic and inorganic substances which are here collectively referred to as *impurities*. The presence of this non sucrose substances is highly detrimental for the process, so the purpose of the next stage of purification is precisely to increase the purity of the process stream as much as possible, by removing, at least, those substances considered more prejudicial to the quality of the final product or to the technological performance of the factory.

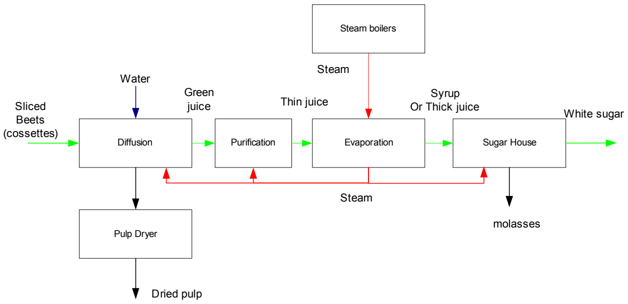


Figure 1. Beet Sugar Factory.

The output of purification, the so called thin juice is then processed in a cascade of evaporators with the purpose of removing an amount of water of about 60% of the juice original mass. The evaporators station has a very efficient design that uses steam coming from the boilers in the first step and then reutilizes the energy of the vapours released by the very water took out form the juice for heating the next steps down the cascade. This arrangement is even capable of providing steam for some other consumers in the factory.

The concentrated syrup is then fed to the Sugar House, whose duty is the making of the final marketable product by means of crystallization. Obviously, the worth of the final outcome of the factory, the white sugar, depends on the correct technological practice of the whole plant, but even a smooth supply of syrup of the right concentration and purity to the Sugar End, could result in white sugar of poor quality by an unwise operation of this last section.

In spite of a continuous automation and instrumentation investment effort on beet sugar factories, the conduction of the whole process still depends heavily on the experience and expertise of the technical personnel. This statement is even more true in the Sugar House: a section that hosts many units of the batch type performing the no so well known process of crystallization of sucrose which depends on the complex and non linear relation of many variables, some of them not directly measured in the typical industrial installation.

In this paper we present a library of components designed for the modelling of the Sugar House. The library has been developed in the modelling language of EcosimPro, a state of the art object oriented simulation tool. The models have been designed with a considerable level of detail and the intended purpose of the library is to make possible the creation of a simulation application for the training of control room operators.

In the following sections of the paper a general description of sucrose crystallization and of the Sugar House process is briefly presented. It then follows a discussion of the main concepts of Object Oriented (OO) dynamic modelling to immediately explain the approach taken for the OO modelling of the Sugar End department. Finally, the use of the library is illustrated by describing with some detail one of the subsections with the presentation of some simulation results.

2. THE PROCESS OF CRYSTALLIZATION.

Crystallization (and its counterpart dissolution) are the more distinctive chemical engineering process taking part in this plant. The driving force of the process of crystallization is the existence of super saturation (***S***) of the solute in the solution i.e. the presence of a concentration of sucrose that exceeds the concentration that defines the solubility of that substance at a given temperature. So, for a pure solution of sucrose:

$$s = \frac{q_{sac/W}}{q_{sat,p}(T)} \qquad (1)$$

Crystallization may happen in a supersaturated solution (s > 1), but it is altogether impossible in under saturated ones. Even more, if crystals are placed in a solution whose super saturation is below unity, they will eventually dissolve.

The conditions of super saturation can be created either by increasing the concentration of sucrose as a result of evaporating water or by decreasing the temperature (see figure 2).

The supersaturated region can be considered as divided in two zones by a more or less fuzzy frontier: there is a metaestable zone where existing sugar crystal can grow up but where the rate of spontaneous nucleation is negligible and the labile zone, in which the probability of nucleation, that is the creation of stable crystal nuclei out of no particulate matter, increases radically.

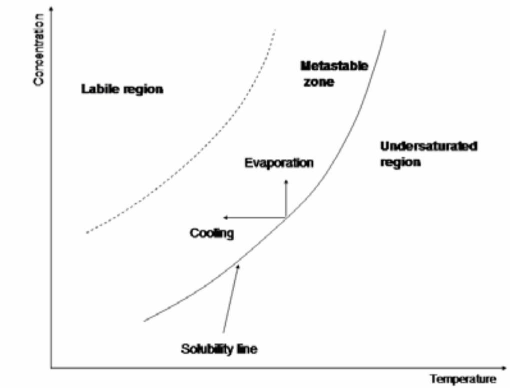


Figure 2. Solubility Curve.

The presence of impurities in the technical solution, raises the solubility of sucrose by a factor ***y_{sat}***. This factor depends no only on the concentration of impurities but also on the internal composition of the different non sucrose substances grouped under this label.

$$s = \frac{q_{sac/W}}{q_{sat,p}(T)\cdot y_{sat}} \qquad (2)$$

The fact that the solubility of sucrose increases with the concentration of impurities means that the crystallization process is impaired by their presence. So the purity of the syrups (as defined in 3) is an indicator that must not be neglected.

$$Pur = \frac{m_{suc}}{m_{suc} + m_{imp}} \cdot 100 \qquad (3)$$

The existence of super saturation is a precondition for the growing of crystals. It is also, the single more important factor in determining the kinetics of the process, but is not the only one. Crystallization is a very complex phenomenon not completely understood (Mullin 2001), it is thought however that it may be explained as the series connection of two concurrent mechanisms: one of mass diffusion from the solution to the crystal surface through a thin stagnant layer, a one of reaction in the surface that models the incorporation of the growing unit to the crystal lattice. The overall rate of the crystal growing process is then mainly determined by the slower of the two mechanisms. The surface reaction step is dependent on temperature and for the case of sucrose this dependence is such that at relatively high temperatures of about 70-80 ºC, the mass diffusion step is the rate limiting one whereas at lower temperatures values the situation is reversed.

So it can be stated that the overall rate of mass transfer from the solution to the crystals depends on many factors: concentration of the sucrose, purity which determines the solubility but also in the sense that some non sucrose substances interferes directly with the incorporation of sucrose molecules to the lattice, temperature through its influence on solubility but also on the surface reaction step, the viscosity and degree of stirring of the mixture via their impact on the mass transfer step.

$$R_g = f(s, Pur, T, \eta, P_{mot}) \qquad (4)$$

The preferred mathematical framework for modeling the kind of particulate systems represented by crystallization process is the one provided by population balance equations (PBE). In this setup, the time evolution of the distribution of some relevant property of the grains, for example a characteristic length (***l***), is tracked by a partial differential equation. For the case of an homogeneous system, a very general PBE could be stated as in (5).

$$\frac{dn(l,t)}{dt} = \frac{\partial[Gn(l,t)]}{\partial l} + n_{in} + B_{nuc} \qquad (5)$$

Where ***n(l,t)*** represents the time dependent density function describing the amount of particles for unit volume that can be found with some specific characteristic length at any given time.

In (5) ***G*** represents the growth velocity of the linear dimension, $\boldsymbol{B_{nuc}}$ the rate of appearance of new particles by nucleation and n_{in} which could be of any sign, the net rate of introduction (or disappearance) of particles in the system due to movement through the boundaries or to breakage or agglomeration of previously existing ones.

The general PBE can be solved numerically to give the shape of the distribution of sizes (the crystal size distribution or CSD as known in crystallization) as the systems evolves. In many applications, however, all that is needed is the knowledge of some of its moments defined as:

$$\mu_k = \int_0^\infty l^k n(l,t) dl \qquad (6)$$

The application of the moments transformation to each term of (5) converts de original PDE in a set of ordinary differential equations (ODE) more easily solvable by standard numerical methods.

The first moments have a definite physical interpretation. So for example μ0 represents the total number of particles, while μ_1, μ_2, μ_3 corresponds to the aggregate value of the length, the surface area and the volume of the grains respectively. From the moments it also easy to obtain other important summary information as the mass average size (L_{ave}) and the normalized standard deviation or coefficient of variation (CV) that gives an indication of the spread of sizes of the CSD.

The complete description of the crystallization systems, must include the relation between the PBE and the continuous part represented by the mass and energy balances. This link must contemplate that the total mass of sucrose that leaves the liquid phase to be deposited over the surface of the growing crystals depends on the conditions of the continuous system and on the total surface of those crystals:

$$W_{cris} = R_g \cdot A_{cris} \qquad (7)$$

But the total area of crystallized sugar, can be obtained directly of the second moment of the CSD.

$$A_{cris} = k_a \mu_2 V \qquad (8)$$

Where $\boldsymbol{k_a}$ is surface factor of the sugar grains that relates the area to the square of the characteristic dimension.

The linear growing rate is, obviously, related to the mass transfer rate which is the ultimate reason of that very increase in the size of crystals:

$$G_c = \frac{k_a \cdot R_g}{3 \cdot k_v \cdot \rho_{cris}} \qquad (9)$$

In (11) $\boldsymbol{\rho_{cris}}$ is the density of the crystals, while ***kv*** is again a shape factor, in this case relating the cube of the linear dimension with the volume of the sugar grain.

3. SUGAR HOUSE DESCRIPTION.

The Sugar House (Van der Poel, Schiweck and Schwartz 1998) receives a concentrated sucrose syrup which has been extracted from beets, and conveniently treated in the previous sections of the sugar factory and delivers the white crystallized commercial sugar.

The growing of the sugar grains is carried out in different types of crystallizers: those where super saturation, the driving force for crystallization, is created by evaporation of water: like the batch operated *tachas* or the continuous crystallizers, or those where it is generated by the controlled reduction of the temperature: cooling crystallizers.

The white sugar must comply with strict quality requirements concerning its purity and the parameters describing the CSD: the population must have the right mean size and ought to be not too spread. The need to meet these quality indicators determines the utilization of batch type crystallizers for growing the commercial grain, because with this type of equipment is possible to exercise a greater degree of control over the process. This is so because in the batch *tachas* it is possible, to avoid a too wide dispersion of sizes by initially seeding the crystallizer with a careful prepared mass of tiny crystals with a narrow CSD. This population is then made to grow along the batch by keeping the super saturation in the safe metastable zone with the purpose of avoiding the creation of new crystals by spontaneous nucleation (Mazaeda and Prada 2007). So, in theory, if the process is correctly conducted, an the end of the strike the same number of originally seeded grains would be obtained with the same standard deviation in its CSD but with a much higher mean size. Of course, the final actual CSD would be wider, because of the inevitable differences in growing speed provoked, for example, by local super saturation fluctuations. In any case, the CSD currently attainable by the much simpler continuous crystallizers, is worse due to intrinsic dispersion of residence times of that mode of operation.

The array of *tachas* delivers a slurry, the so called massecuite, where the sugar crystals are embedded in the original syrup, now called mother liquor. In order to obtain the final product, the sugar grains need to be separated from the surrounding syrup, and this task is performed in an arrangement of centrifuges. The centrifuges, in this case, are also of the batch type, and for a similar motivation of quality. The syrup obtained just after loading the rotating basket with the slurry, is essentially the separated mother liquor. But it is impossible to completely eliminate the syrup from the crystals by using only the mechanical separation power. So to comply with the purity standards of the final product, at some point in the centrifuge cycle, it is imperative to inject clear water in the basket to extract, by diffusion, and push out of the crystal bed the traces of mother liquor adhered to the surface of the grains. The water extracts the remaining liquor but also dissolves part of the solid sugar, so the expelled liquid has now a greater level of purity. Hence, the syrup from the centrifuges is divided in two streams of different purities: poor syrup first and then rich syrup. The wet sugar crystals are finally discharged to be transported to a dryer for removing water and then to the appropriate silos and out to market.

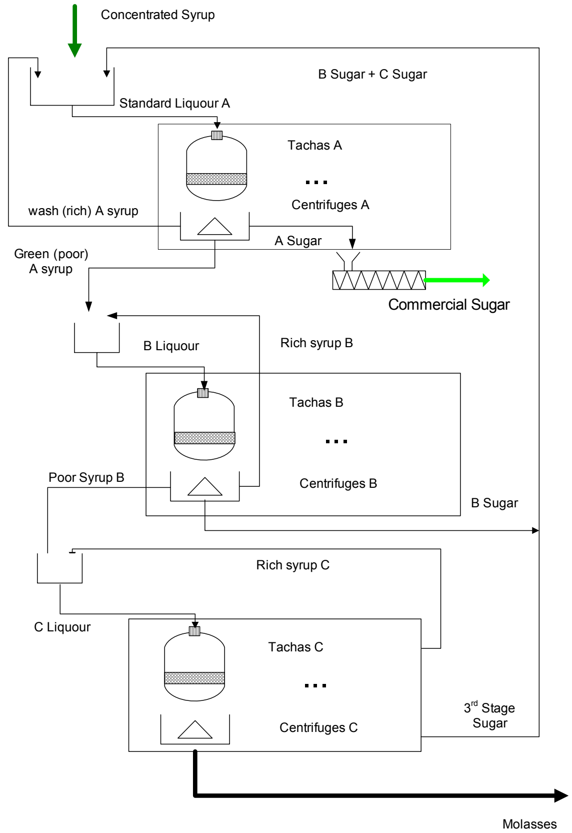


Figure 3. Sugar House Layout.

In relation to the syrup streams, it is economically justified to recover the extra dissolved sucrose by recycling the rich one back to be fed to the crystallizer mixed with incoming concentrated juice. The poor syrup, on the other hand, which contains almost all the impurities put in the department, if recycled would compromise the process by lowering the purity of the standard liquor, without bringing in any benefits. But, it can't just be throw away either, because it still contains too much dissolved sucrose: the strategy adopted in most sugar houses is to continue the processing of poor syrups in subsequent stages of similar structure to the one just described.

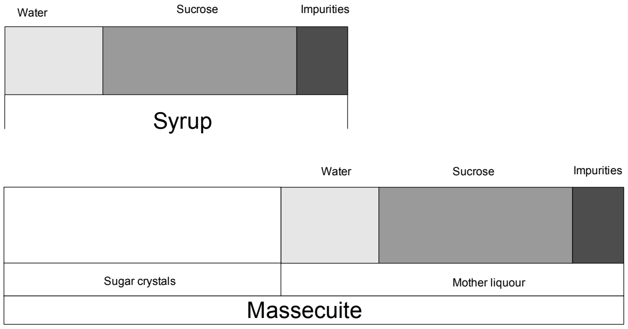


Figure 4. Syrup and Massecuite Composition.

Most modern Sugar Houses employ two more stages: know as B and C stages. The functioning is very similar to the already explained first product white sugar or A stage. But the objective now of crystallization is the separation of as much sugar as possible in order to recover it for the formation of first product. These two extra exhausting passes don't have an immediate concern on the quality of the their produced sugar as a market product, so the requirements of a narrow CSD is somehow relaxed but not too much: a sugar with mixed grain or with a large amount of fines complicates the process of separation in the centrifuges.

The B and C stages have to the deal with the problem of the increasing viscosity of the process streams. The syrups have now a low (and decreasing) purity, and this fact implies a higher level of concentration to achieve the needed super saturation: thus the increased consistency of the mass. This fact makes that the moving around of syrups becomes a significant problem which affects the selection of the processing units: nonetheless in the B stage it is still possible to use batch *tachas* for crystallization but the centrifuges must be of the continuous type. In the C stage, almost all crystallization equipment is of the continuous kind: at first evaporative crystallizers are used but the final exhaustion of masses is carried out in a cascade of cooling crystallizers.

On the other hand, the speed of crystallization is reduced considerably: so that if a strike of first product takes around two hours, the residence time of the crystallization towers installation at the C stage is in the order of the 45 hours.

The poor syrup of the third product, known as molasses, with a purity of around 60%, contain almost all the original impurities but also that part of sucrose that cannot be efficiently recovered with the standard industrial equipment. Molasses are thus a final byproduct of the Sugar End that can be used for the production of alcohol.

4. OBJECT ORIENTED DYNAMIC MODELLING SUGAR HOUSE LIBRARY.

The Object Oriented (OO) paradigm, originally introduced in the context of general purpose software design, had been extended to the modelling of dynamic systems.

The object orientation, in this context, provides the modellers with the language constructs that let them manage complexity by allowing the creation of reusable components. This approach allows what is called *physical modelling,* because it encourages the creation of large models by interconnecting the different pieces just in the same manner in which a physical world prototype would be constructed. The models so designed could in its turn be made available as components to a higher level of abstraction, making possible in this way the successful hierarchical decomposition of otherwise very difficult problems.

The object concept implies encapsulation which means that the internal structure and behaviour is hidden from the rest of the model. The interaction between components is carried out through the so called *ports*, which are special objects, also defined by the modeller, used to specify the variables to be interchanged and the mathematical behaviour of those variables at the junction.

For the object approach to work, the differential- algebraic (DAE) mathematical equations that describe them, must be stated in an non causal, declarative way (Cellier and Kofman 2006). Once the final, aggregated model is defined, the computational causality can be established, the boundary variables are identified and the model is ready to be exercised by performing experiments on it.

The advantages of OO modelling have been put to use in the creation of the Sugar House library.

The library is a collection of class models representing the main processing units already mentioned. All the classes are comprehensively parameterized and coded in a general way so that when used several times to create an specific flowsheet configuration, each instantiated object can receive its detailed set of parameters exactly defining its physical and operational characteristics.

The main components of the library are:

- **Batch evaporative crystallizer (*tacha*)**: a complex class which codifies a model of a discrete-continuous hybrid nature. The continuous dynamical model expresses the material, energy and population balances describing the physics of the particulate system. The crystallizer is represented by a closed container where the massecuite is processed and where super saturation is created by means of the evaporation of water in vacuum conditions and the controlled feeding of syrup. The evaporation process is conducted at low pressures in order to be able to work at correspondingly low temperatures to avoid the caramelization or thermal degradation of sucrose. The heat needed to sustain the evaporation of water is provided by a tube and shell heat exchanger or calandria located in the lower half of unit: the mass inside the crystallizer moves inside the tubes and the heat is transferred by virtue of the condensing steam in the shell. The mass inside is vigorously stirred by a mechanical agitator guaranteeing its homogeneity. The stirrer has also an important role in increasing the kinetics of crystallization and on enhancing the coefficient of heat transmission between the mass and the calandria. The model also incorporates the program of the crystallizer implemented by a complex fully parameterized state machine that drives the stage by stage evolution of the strike. The typical stage sequencing includes: the initial loading with feeding syrup, a concentration by evaporation phase after establishing a vacuum pressure in the vapor phase space on top of the liquid, the introduction of the crystal seed, the growing of the sugar crystals, the final conditioning of the consistency of the slurry, the discharge of the massecuite, and finally a cleaning stage with steam to remove any traces of products and to prepare the unit for the next strike. The class representing the *tacha* can be instantiated as many times as needed, and each particular object can have its own defining parameters. The behavior of the object depends of the characteristics of the incoming syrup: so each one will reproduce the very different performances of the real units when connected in the different stages (A, B or C) of the plant.
- **Batch centrifuge**: It is also an hybrid model which describes the separation of the mother liquor from the sugar crystals provoked by the centrifugal force produced in a perforated rotating basket. A program guides the operation of the equipment in each step of the cycle which includes: the loading of massecuite with the simultaneous initial separation of mother liquor, acceleration of the basket and injection of clear water for removal of syrup traces, the slowing down of the drum and the final discharge of sugar crystals. The component describes the processes of filtration of the mother liquor and water through the developing sugar bed. The dissolution of sugar crystals is also modelled when super saturation falls below unity during water washing. The dynamics of bed formation is reflected and the dependence of bed filterability on the characteristics of the CSD of sugar is accounted for. The configuring parameters of the class include: geometrical dimensions of the basket and its angular speed profile. The timing of the different stages of centrifuge program can be freely specified. It is specially important the opportunity of establishing the poor to rich syrup switching time: a key parameter used by the technical personnel to achieve an efficient operation of department.
- **Continuous centrifuges**: It also separates the sugar crystals from the mother liquor. The process in this case is continuous and this fact makes possible the handling of more viscous masses but, on the down-

side, it cannot achieve the same degree of purity of the extracted crystals. This characteristics make this kind of centrifuges the right choice for B and C stages. This unit can be provided in two version: with and without separation of low and high purity syrups.

- **Continuous evaporative crystallizer**: The massecuite travels from input to output through a series of compartments. The super saturation is maintained in the metastable zone inside each cubicle by balancing the amount of water evaporated at vacuum conditions with the introduction of the right amount of syrup. The heat needed for evaporation is provided by means of the condensation of steam flowing inside a bundle of tubes arranged in the longitudinal direction of the equipment.
- **Cooling crystallization tower**: It is essentially a heat exchanger where the now very viscous slurry is made to circulate around an array of tubes conveying cold water in a counter current way. The super saturation is increased by virtue of its dependence on temperature. The main technological difficulty with this kind of equipment is the very high viscosity of the magma, and this problem is made even worse by the reduction of temperature. So, a cooling tower can be configured in such a way as to be able to provide for the heating of the massecuite in its final stretch by allowing the circulation of hot water in a separated circuit. Care must be exercised, however, to achieve the desired reduction of viscosity avoiding the dissolution of crystals by heating just to the saturation point but not lower.

In both type of continuous crystallizers is important to represent correctly the retention time of the real equipment because of its relation with the width of the resulting CSD. This is accomplished by the series connection of number of sections modelled as ideally mixed stirred tanks.

In all the mentioned units models the dynamics of crystal growth is represented by the evolution of the moments of number density function. The use of the moments transformation as a model reduction technique is imperative to alleviate the numerical load because the intended purpose of the model library is the creation of large simulated flowsheet for the entire department.

It has been considered relevant to model up to six moments of the PBE given the existence of expressions that describe the rate of some phenomena, for example nucleation, as depending on μ_5.

The above described main units of the house, can be instantiated several times, in each case specifying a different set of configuring parameters, to describe the precise disposition of the concrete plant. But in any complex process industry there is a collection of other indispensable standard auxiliary equipment like, for example, buffer tanks, heat exchangers, pumps, valves and even controlling devices like PIDs. Configurable classes representing the mentioned standard apparatus have been developed as part of a common library designed to assist in the modelling of the complete sugar factory (Merino, Mazaeda, Alves, Acebes and Prada 2006]. However, this standard equipment is not only helpful in describing the auxiliary elements surrounding the main units, but it is also used as building blocks for the internal description of those very major units. In this way, taking advantage of the hierarchical composition power of object orientation, a complex component, like the one representing the *tachas*, is made up of an assortment of lower level components like valves and PIDs.

The connection of the different models requires also the premeditated design of the appropriate *ports*. Some of these are provided as part of the already mentioned common library, as for example, the one representing the flow of saturated or super heated steam. Others are defined as part of Sugar House Library like those describing massecuite and syrups.

In all cases, the ports should introduce the needed equations to explain the behaviour of typical process variables like pressure and total mass flow. To describe the conduction of pressure for example, which is a variable of type *across* (in bond graph terminology), the port coding must indicate the need to add an extra restriction equation for each junction specifying the equality of the corresponding variables in all the converging branches. On the other hand, variables of the *through* kind, like mass flow, calls for the introduction of an equation stating that the sum of the flows at the node must be cero. The inclusion of the mentioned restriction equations for each node in the final model can be as simple as using the appropriate modelling language constructs, see for example figure (5) and eq. (10).

```
PORT syrup
     EQUAL  REAL P
     SUM    REAL W

     EQUAL OUT REAL h
     SUM   IN  REAL H
     …
CONTINUOUS
     H=W*h
END PORT
```

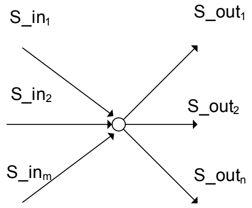


Figure 5. Syrup Port Definition Fragment.

$$S_1.P = S_2.P = \cdots S_n.P$$
$$\sum_{i=1}^{n} S_i.W = 0 \qquad (10)$$

There are, on the other hand, magnitudes which have a mixed kind of behaviour: the value of the variable of the branches into the node must be summed up, but it must be considered as equal in every departing conduit. This can be observed in transport variables concerning for example the transmission of heat by convection in the direction of the flow: the total enthalpy in the node is the sum of the contribution of all incoming flows, but the specific enthalpy of all the departing nodes is the same:

$$
\begin{aligned}
&H = \sum_{i=1}^{m} S\_in_i.H \\
&S\_out_1.h = \cdots = S\_out_n.h = h \qquad (11) \\
&H = W \cdot h
\end{aligned}
$$

In the case of the port defined for the transport of the massecuite streams, there is the additional need of conveying information relative to the shape of the CSD. For the reason already discussed, it has been decided to pass on the values of the first six moments of the distribution.

Each one of the moments must be considered as a transport variable so each one receives a similar mathematical treatment. For example in figure (6) and equations (12) the procedure is exemplified for the case of the cero moment. The total flow of the incoming quantity (in this cases total number of particles per unit time) are added, while the specific value of all the outgoing branches, in this case the number of particles per unit time per unit volume, is forced to be the same. In (12) Q_{mm} stands for the total volumetric flow rate of the mother liquor of the slurry streams arriving at the node.

```
PORT massecuite
     EQUAL OUT REAL fm0
     SUM   IN  REAL Fm0
     …
CONTINUOUS
     Fm0=Qmm*fm0
END PORT
```

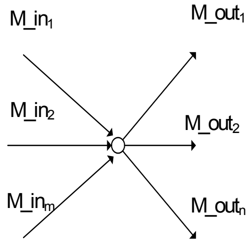


Figure 6. Massecuite Port Definition Fragment. CSD Moments Transport.

$$
\begin{aligned}
&Fm_0 = \sum_{i=1}^{m} M\_in_i.Fm_0 \\
&M\_out_1.f\mu_0 = \cdots = M\_out_n.f\mu_0 = f\mu_0 \qquad (12) \\
&f\mu_0 = {}^{Fm_0}\!\!\diagup_{Q_{mm}}
\end{aligned}
$$

Each one of the components integrated in the sugar house library has their own set of modelling assumptions in tune with the declared operators training purpose of the whole project. The targeted scale of the kind of application tool to be build based on the library, aiming at describing the entire Sugar House plant as part of a scheme including a complete Sugar factory, also determined the degree of resolution attainable when dealing with each individual detail. The rationale followed has been the being able to offer to the would-be trainee a model that represents correctly the main known relations between the many variables involved by scrupulously stating the pertinent balance equations and by reflecting accurately the known structure of each of the units. The exact representation of the external configuration of the specific sugar house flowsheet depends of each particular example but in any case is made possible by the very object oriented nature of the library. Additionally, the reproduction of the programs of the batch units like the *tachas* and first product centrifuges has been very realistic and detailed.

A more difficult issue is the truthful accounting of the rate processes like for example, the kinetics of crystallization and dissolution or the heat transfer coefficients. Here the challenge is not only the accurate reproduction of the observed dynamic behaviour but to make that behaviour a meaningful function of the relevant prevailing parameters and properties of the concerned streams. So for example, it is known that the heat transfer coefficient is not more than a convenient simplification of a more complex physical phenomenon, and for this reason cannot be simply approximated by a fixed number suitable for every situation. The accepted practice in the chemical engineering community is the approximation of this kind of rate coefficients by experimental correlations between the appropriate non dimensional numbers adapted to the specific situation. For example, the heat transfer coefficient can be typically obtained as function of the Reynolds, Prandtl and Nusselt numbers and this gives the chance of putting the referred parameter in relation with the prevailing conditions of the flow, the thermal properties of the fluid, and geometry of the problem. Similar relations do exist to obtain the mass transfer coefficient which is important in describing the velocity of crystallization and dissolution. The approach here taken is to make use of the existing experimental or heuristic relations for describing the mentioned dynamics in the major units, especially those of batch kind where the conditions are widely modified along the strike. In any case, there is always room for adapting free parameters of the models to match the available measured variables values obtained from the specific real plant.

Another important concern relates to the modelling of the mixing behaviour of the different types of equipment. For batch units it is somehow sustainable the claim of a perfectly mixed tank which translates into a simple, global parameter ordinary differential equations model. A Continuous crystallizer, on the other hand, exhibits a more complex residence time distribution (RTD) function, halfway between the two ideals patterns of perfectly mixed reactor and plug flow reactor. Since the expected characteristics of the final CSD, in particular its width, depends critically on the residence time of the sugar crystals in the reactor, it is important to get right this behaviour. The approach here followed has been to approximate the RTD by placing a cascade of an enough number of perfectly mixed reactor models.

To model auxiliary equipment like heat exchangers correctly, it would be necessary to follow a similar discretization strategy. Although perfectly possible, the approach here suggested is to pragmatically solve the trade off between dynamic accuracy and easiness of the numerical solution in favour of the latter.

Other general important modelling assumptions of the library have already been mentioned in passing, but we explain them here giving more details for the sake of precision: all the non sugars present in the technical streams are grouped under the impurity tag. The exact composition of the impurities is of course important, but it would be extremely difficult to keep track of all these substances individually. In any case, by considering all non sugars together, we here follow the common practice of most sugar industry studies. In those cases in which the exact nature of non sugar substances has a big impact, like for example when describing their influence on solubility, the unpredictability caused by the ignorance of the exact impurities composition is accounted for by means of suitable adjustable parameters. Furthermore, the library do not contemplate either the possibility of mass transfer between the impurities and any of the other species integrating the syrups or the massecuite. This, for example, rule out the considering of the process of inversion, not excessively important in the sugar house anyway, whereby the sucrose molecules are transformed in inverted sugars which are termed as impurities. The process of impurities inclusion in the sugar crystals is not modelled either: this phenomenon can be observed when the crystals grow up too rapidly and traces of mother liquor gets trapped inside the lattice.

Finally it should be said that the library must use the relevant properties of the described streams like specific enthalpies, densities, viscosities and the like. The properties of steam are widely known but those specifically pertaining to the sugar industry have been taken from the excellent compilation due to Bubnik, Kadlec, Urban y Bruhns (1995).

5. THE MODEL OF THE SUBSECTION OF STANDARD LIQUOR FORMATION.

In order to better illustrate the concepts involved in the design and use of the library, the subsection in charge of the formation of standard liquor for the A stage is described. This part of the Sugar House (fig. 7), whose purpose is to provide the feeding liquor to the first stage crystallizers, consist of a series of continuously operated stirred tanks which are fed with the concentrated syrup entering the department and with the rich syrups from the A centrifuges. The tanks also receive the sugars from the B and C stages, and the goal is to completely dissolve all the incoming crystals. The temperature of the equipment is kept in a safe high value to help the dissolution process by increasing the solubility of the solution. The concentration of the resulting liquor is automatically controlled by manipulating the input of a more diluted juice.

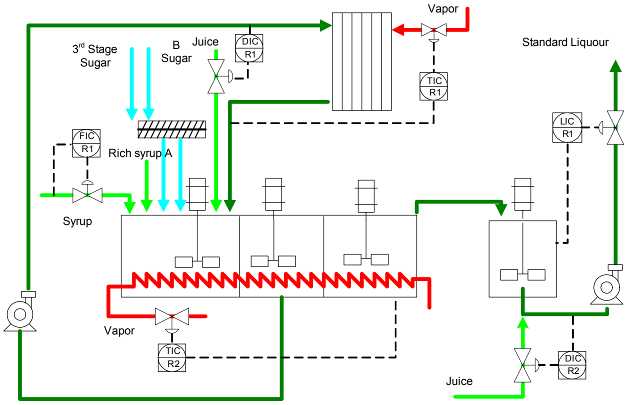


Figure 7. Standard Liquor Formation.

In figure 8 an EcosimPro (EcoDiagram) connection diagram model describing the central, most important part of the subsection is shown. The aggregated model incorporates readily a few instances of model classes representing standard equipment such as valves, pumps, heat exchangers and PID controllers. A side by side inspection of the section flowsheet representation (fig.7) and its model assembled graphically by reusable object models (fig. 8) gives an idea of how intuitive the OO way of modeling could be.

The class ***compartment*** has been specifically developed for this section. It represents a perfectly mixed control volume that is able to receive the two mentioned types of streams: syrup and massecuite, and which delivers, generally by overflowing, the resulting slurry. The main process being described in the component is the dissolution of the existing sugar crystals.

The other especial class used in the final model is the one representing a steam coil used for the heating of the mass inside each of the cubicles.

The original processing tank is build using an elongated horizontal open reservoir, internally divided in three compartments by partition walls. In the model this fact is represented by placing the same number of instances of the class ***compartment*** (*comp1* to *comp3*) in series. In this example, it has been assumed that each of the original compartments behaves as a perfectly mixed reactor. Even though each real partition has its own dedicated stirrer, the previous claim might no be entirely justified. In any case, if RTD data were available, it would be straightforward to approximate the corresponding mixing behavior, by placing the appropriate number of ***compartment*** objects in cascade.

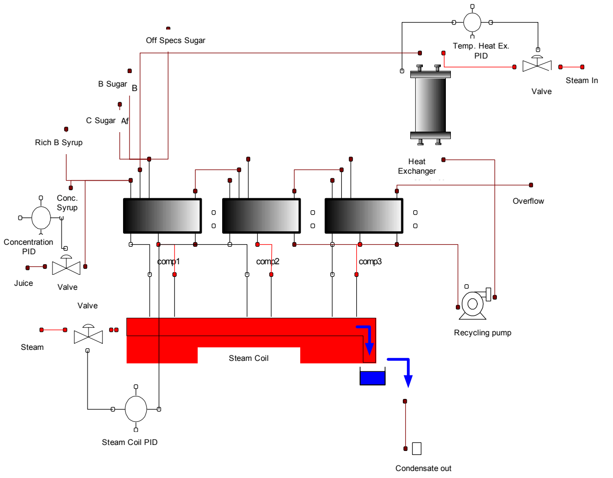


Figure 8. EcosimPro Internal Description of the Horizontal Dissolution Tank.

The mathematical model of each compartment keeps track of the presence of each of the constituents of the slurry by stating the corresponding mass balances.

The most important element to consider here is the rate at which mass is transferred from the dissolving crystals to the solution. This rate in [kg/s] can be determined according to:

$$W_{dis} = k_d A_{cris} C_{sug} (s-1) \qquad (13)$$

Dissolution can be considered as the counterpart of crystallization, but in this case, the phenomenon can be successfully described by just one mechanism: mass diffusion from the crystals to the bulk of the solution: so k_d is the mass transfer coefficient [m/s], C_{sug} the concentration of sucrose in a saturated technical solution [kg/m^3], A_{cris} the total area [m^2] of existing crystals obtained from PBE and S the supersaturation which should be less that unity for dissolution to occur.

The mass transfer coefficient can be adjusted in accordance to the measurements, or, in the absence of observed data it could be approximated by correlations between non dimensional numbers as, for example, in Koiranen, Kilpio, Nurmi and Norden (1999).

The energy balance is also written down in the model to correctly describe the influence of temperature on the workings of the installation. Figure 9 illustrates the behavior of the compartments in cascade in relation to the temperature of the mass. The graph depicts the evolution of the sugar mass fraction along with the value of supersaturation, which is the driving force of the dissolution process and whose dependence on temperature is the ultimate cause of the above mentioned correlation. It can be observed, that for a temperature of 65ºC, the sugar content decreases form a compartment to the next, but the equipment is unable to dissolve all the incoming grains. When the temperature of the mass is increased by modifying the set point that controls the steam input to the heating coil, initially to 80 ºC and then to the normal working value of 90ºC, the super saturation is correspondingly lowered and the sugar content goes to zero first in the last cubicle and then, in the intermediate one also. A high temperature helps the dissolution process, but values above 90ºC should be avoided to prevent *caramelization*.

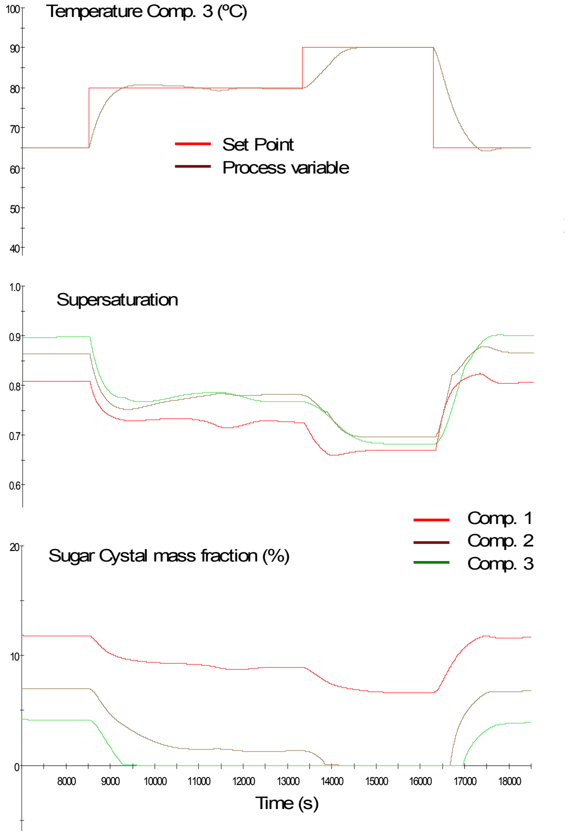


Figure 9. Influence of Temperature on Dissolution.

There are many other factors determining the performance of the subsection: figure 10, for example illustrates the impact on the performance of the equipment of the amount of sugar crystals originally fed to the first compartment.

The control room operator should be trained in the not so obvious relationships of the different variables to be able to take the appropriate actions when confronted with the real factory challenges. It may very well happen, for example, that the quality of the available steam is incapable of guaranteeing the recommend mass temperature, she might then try to compensate this effect by a suitable manipulation of the proportion between the different streams put in.

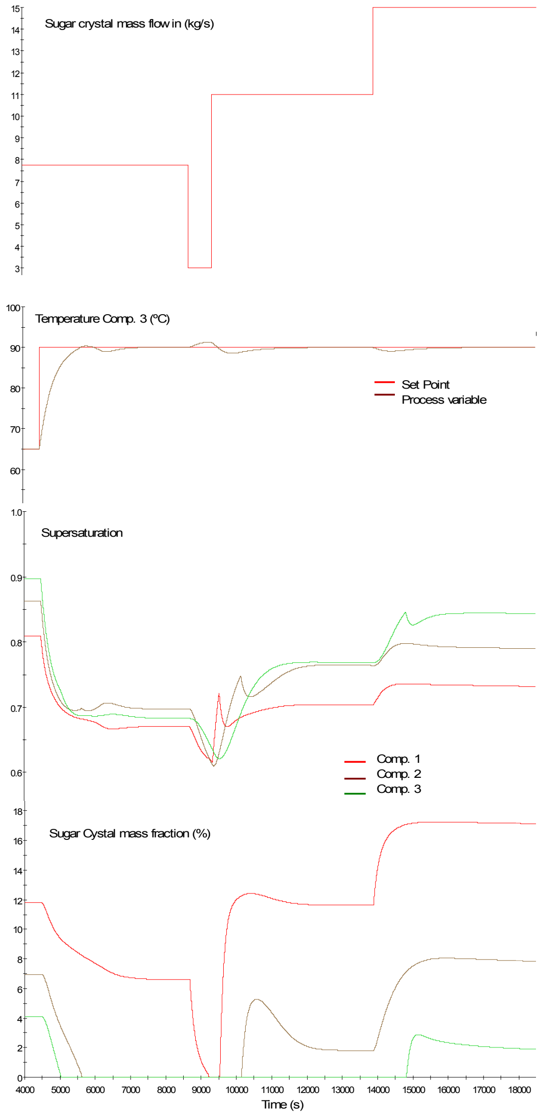


Figure 10. Sugar Crystals Mass Fraction in each Compartment with Different Mass Flows of B Sugar.

Other elements should be of concern in the conduction of the subsection. It is important, for instance, the guaranteeing of a constant purity in the liquor sent to the downstream *tachas*. So if the purity of the arriving concentrated syrup changes, the operator should counterbalance this fact by adjusting the amount of rich syrup introduced in the dissolution tank.

Finally, it should be said, that for a useful training experience, the developed dynamical models should be presented to the trainee with an HMI interface (fig. 11) that resembles the one he would meet in the real factory. The training application (Alves, Normey-Rico, Merino, Acebes and Prada 2005) also guarantees the real time execution of the simulated sections.

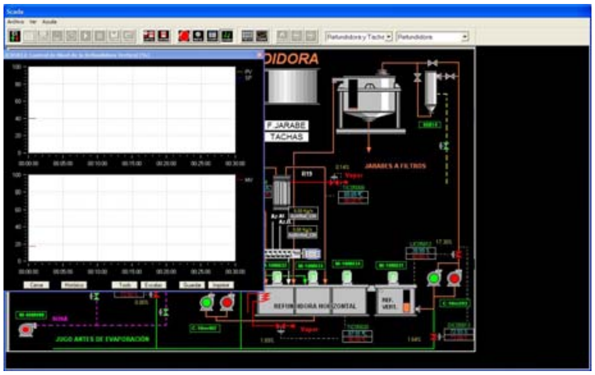


Figure 11 HMI Interface of the Training Tool.

ACKNOWLEDGMENTS

The authors would like to express their gratitude to Azucarera Ebro S.L. (Spain) for their financial support and technical assistance throughout the development of the Sugar Factory Dynamic Simulator and to the Ministry of Science and Education of Spain for their financial support through the project PTR1995-0820-OP.

REFERENCES

Alves, R., Normey-Rico, J.E., Merino, A., Acebes, L.F., Prada, C. de, 2005. OPC based distributed real time simulation of complex continuous processes. *Simulation Modelling Practice and Theory*. (13)7, 525-549.

Bubnik, Z., Kadlec, P., Urban, D., Bruhns, M., 1995. *Sugar Technologist Manual. Chemical and Physical Data Manufacturers and Users*. Germany: Verlag Dr. Albert Bartens.

Cellier, F.E., Kofman, E., 2006. *Continuous Systems Simulation*, New York: Springer-Verlag.

Koiranen, T., Kilpio, T., Nurmi, J., Norden, H.V., 1999. The modeling and simulation of dissolution of sucrose crystals. *Journal of Crystal Growth*. (198/199), 749 - 753.

Mazaeda, R., Prada, C. de, 2007. Dynamic Simulation of a Sucrose Batch Evaporative Crystallizer for Operators Training, *Proceedings of the 19th European Modeling and Simulation Symposium (Simulation in Industry), EMSS 2007*, Bergeggi, Italia.

Merino, A., Mazaeda, R., Alves, R., Acebes, L.F., Prada C. de, 2006. Sugar factory simulator for operators training. *Proceedings of the 7th IFAC Symposium in Advances in Control Education: ACE 2006*, Madrid, Spain.

Mullin, J.W., 2001. *Crystallization*. 4th Edition London: Butterwoth-Heinemann.

Van der Poel, P.W., Schiweck, H., Schwartz, T., 1998. *Sugar Technology. Beet and Cane Sugar Manufacture*. Germany: Verlag Dr. Albert Bartens.

ANALYSIS AND OPTIMIZATION OF COMPLEX SMALL-LOT PRODUCTION IN NEW MANUFACTURING FACILITIES BASED ON DISCRETE SIMULATION

Doc. Václav Votava, CSc.[(a)], **Ing. Zdeněk Ulrych, Ph.D.**[(b)], **Ing. Milan Edl, Ph.D.**[(c)],
Ing. Michal Korecký, Ph.D.[(d)], **Ing. Václav Trkovský, CSc.**[(e)]

[(a) - (c)] Department of Industrial Engineering and Management, University of West Bohemia in Pilsen, Univerzitni 8, Pilsen, Czech Republic
[(d) - (e)] SKODA HOLDING a.s., Tylova 57, Pilsen, Czech Republic

[(a)]votava@kpv.zcu.cz, [(b)]ulrychz@kpv.zcu.cz, [(c)] edl@kpv.zcu.cz
[(d)] michal.korecky@skoda.cz, [(e)] vaclav.trkovsky@skoda.cz

ABSTRACT

This paper focuses on simulation study of small-lot production of complicated products with a long technological time. The paper describes a possible procedure for generating simulation models based on incomplete data from EIS. This simulation model must flexibly react to changes in complicated technological steps. This paper describes one possible solution to this problem.

Keywords: small-lot production, discrete simulation, simulation system ARENA

1. INTRODUCTION

All companies now use Information Systems (IS) for planning and controlling their businesses. Basic information saved in an IS is used to support the controlling of manufacturing processes in companies for each type of product. When a company wants to transfer production to a new manufacturing space with new manufacturing facilities, it is appropriate to verify project documentation, new technology and all static capacity calculations by dynamic calculation – simulation. On the basis of the experiments which were carried out, it is possible to undertake organizational precautions which are essential before starting new production.

This paper focuses on the creation of a simulation model. This simulation model must flexibly react to changes in complicated technological steps. The method used must preferably be simple and fast. This paper describes one possible solution to this problem. The solution deals with a simulation study of small-lot production of complicated products with a long technological time for the company ŠKODA TRANSPORTATION. The verified products include:

- Production of locomotives,
- Production of trams,
- Renovation of metro wagons,
- etc.

A simulation model is created which enables very simple adaptations to various types of products with various manufacturing plans.

2. METHODOLOGY

This section describes the methodology used in the simulation study. The whole approach is illustrated in Figure 1.

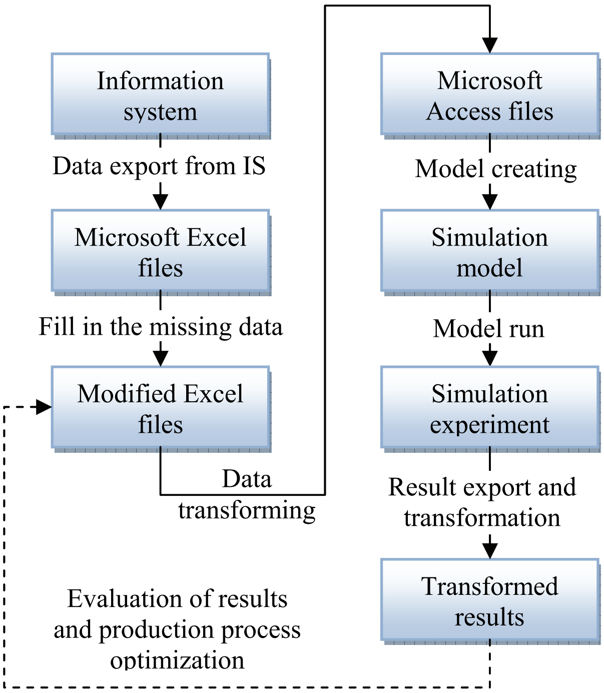


Figure 1 Methodology of the presented simulation study

From Figure 1 we can see that the whole process of the simulation study is divided into several basic steps. These steps are:

- Data export from IS into MS Excel Files,

- Filling in the missing data (e.g. transport between workplaces),
- Data transformation to MS Access files suitable for simulation model,
- Simulation model creating,
- Model run,
- Simulation results export and transformation into MS Excel or MS Project Files
- Evaluation of results and production process optimization.

The above mentioned steps are using the components in boxes in Figure 1:

- Information system,
- Microsoft Excel files,
- Modified Excel files,
- Microsoft Access files,
- Simulation model,
- Simulation experiment,
- Transformed simulation results.

2.1. Information system

An information system is the fundamental data resource for a simulation model. From the information system we obtain the following basic information about each product:

- sequence of production operations,
- technological time of each operation.

2.2. Microsoft Excel files

The ability of many information systems to export data from IS to Microsoft Excel files in a structured form is used here. Data structures have to be the same all the time. This is important for following automatic input data processing.

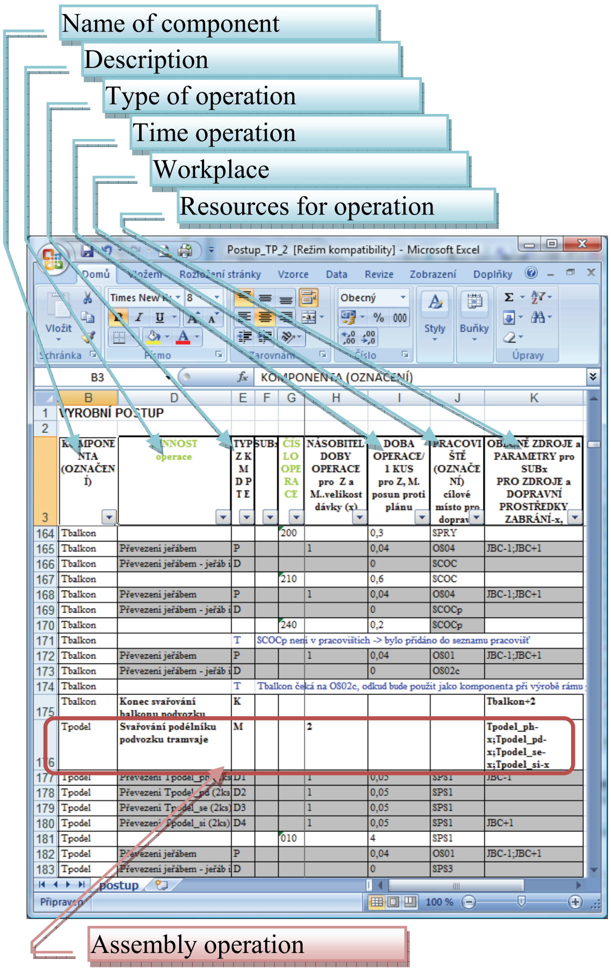
From Figure 1 we can see that MS Excel files are used as a transfer step between the information system and the database which uses the simulation model. This step is important for data retrieval for the simulation model. The simulation model verifies products which have not yet been produced, therefore some products do not yet have a final technological operation described. For this reason we chose MS Excel, because Excel very simply allows modification and filling in of missing input data (estimated or existing). Excel allows very simple and independent preparation of input data for individual experiments.

2.3. Modified Excel files

An information system does not usually contain all the necessary input data for a simulation model. Missing input data must be manually entered in this step. In this step we also reduce the technological process so that outputs from the simulation model are not misrepresented and the simulation runs well and faster. All inputs for the model are saved to several Microsoft Excel files. Individual files contain the following information:

- technological sequence of each product (including hierarchical structure (components) of product) – technological sequence of one product can be divided into more files
- system information
  - list of resources – including basic information about resources
  - list of components - including basic information about components
  - list of transport equipment – including basic information about transport equipment
  - list of workplaces – including basic information about workplaces
- calendar – list of shifts and definition of these shifts. Shifts are defined for all resources, transport equipment and workplaces used
- production plan – this file creates a schedule of the production plan for each product in the simulated time period

The structure of records in the files must have clear rules. These rules are important for following the automated processing of input data.



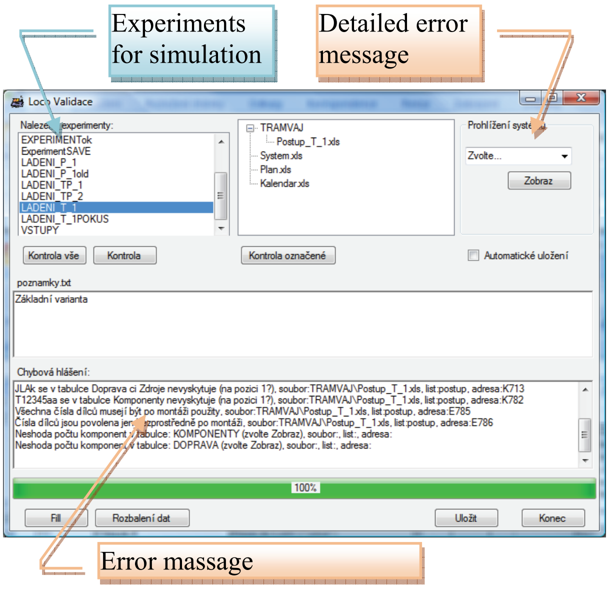
	B	D	E	F	G	H	I	J	K
1	VYROBNÍ POSTUP								
2									
3	KOMPONENTA (OZNAČENÍ)	INNOST operace	TYP Z K M D P T E	SUBx	ČÍSLO OPERACE	NÁSOBITEL DOBY OPERACE pro Z a M..velikost dávky (x)	DOBA OPERACE/ 1 KUS pro Z, M. posun proti plánu	PRACOVIŠTĚ (OZNAČENÍ) cílové místo pro doprav	OBECNÉ ZDROJE a PARAMETRY pro SUBx PRO ZDROJE a DOPRAVNÍ PROSTŘEDKY ZABRÁNÍ-x,
164	Tbalkon				200		0,3	SPRY	
165	Tbalkon	Převezení jeřábem	P			1	0,04	OS04	JBC-1;JBC+1
166	Tbalkon	Převezení jeřábem - jeřáb i	D				0	SCOC	
167	Tbalkon				210		0,6	SCOC	
168	Tbalkon	Převezení jeřábem	P			1	0,04	OS04	JBC-1;JBC+1
169	Tbalkon	Převezení jeřábem - jeřáb i	D				0	SCOCp	
170	Tbalkon				240		0,2	SCOCp	
171	Tbalkon		T	SCOCp není v pracovištích -> bylo přidáno do seznamu pracovišť					
172	Tbalkon	Převezení jeřábem	P			1	0,04	OS01	JBC-1;JBC+1
173	Tbalkon	Převezení jeřábem - jeřáb i	D				0	OS02c	
174	Tbalkon		T	Tbalkon čeká na OS02c, odkud bude použit jako komponenta při výrobě rámu					
175	Tbalkon	**Konec svařování balkonu podvozku**	**K**						**Tbalkon+2**
176	Tpodel	**Svařování podélníku podvozku tramvaje**	**M**			**2**			**Tpodel_ph-x;Tpodel_pd-x;Tpodel_se-x;Tpodel_si-x**
177	Tpodel	Převezení Tpodel_ph ks	D1			1	0,05	SPS1	JBC-1
178	Tpodel	Převezení Tpodel_pd (2ks	D2			1	0,05	SPS1	
179	Tpodel	Převezení Tpodel_se (2ks)	D3			1	0,05	SPS1	
180	Tpodel	Převezení Tpodel_si (2ks)	D4			1	0,05	SPS1	JBC+1
181	Tpodel				010		4	SPS1	
182	Tpodel	Převezení jeřábem	P				0,04	OS01	JBC-1;JBC+1
183	Tpodel	Převezení jeřábem - jeřáb i	D				0	SPS3	

Figure 1 File example - Manufacturing sequence

For example, in Figure 2 is shown an MS Excel file, which describes the production process of one product. This file describes basic information about a production process:

- sequence of production operations
- required resources for production operations
- workplace for production operations
- hierarchical structure of product

In this step we change data by hand, which may result in some errors in the data. This step is followed by formal data control by special software created for this purpose. An example of this control software is shown in Figure 3.



poznamky.txt

Základní varianta

Chybová hlášení:

```
JLAk se v tabulce Doprava ci Zdroje nevyskytuje (na pozici 1?), soubor:TRAMVAJ\Postup_T_1.xls, list:postup, adresa:K713
T12345aa se v tabulce Komponenty nevyskytuje (na pozici 1?), soubor:TRAMVAJ\Postup_T_1.xls, list:postup, adresa:K782
Všechna čísla dílců musejí být po montáži použity, soubor:TRAMVAJ\Postup_T_1.xls, list:postup, adresa:E785
Čísla dílců jsou povolena jen bezprostředně po montáži, soubor:TRAMVAJ\Postup_T_1.xls, list:postup, adresa:E786
Neshoda počtu komponent v tabulce: KOMPONENTY (zvolte Zobraz), soubor:, list:, adresa:
Neshoda počtu komponent v tabulce: DOPRAVA (zvolte Zobraz), soubor:, list:, adresa:
```

Figure 2 GUI Application LocoValid

2.4. Microsoft Access files

Missing information was added to all Excel files in the previous step. Input data must now be transferred from Microsoft Excel files to Microsoft Access files in order to automatically generate the simulation model. This transfer is important to accelerate the work in the simulation model. MS Access is simply supported both by MS Excel and the ARENA simulation system.

2.5. Simulation model

A simulation model is created in the Rockwell Software simulation system, ARENA v. 11. The model contains a variable part and a fixed part.

2.6. Fixed part of model

The fixed part of the model is not modified during any simulation experiments. This part of the model describes the typical behaviour of all workplaces. Simple animation also belongs to this part of the model and shows the position of separate workplaces and transport between workplaces.

The fixed part of the model is permanently prepared beforehand.

2.7. Variable part of model

This part of the model depends on the production process of each product, on setting resources and a production schedule. This means we cannot create this part of the model before we simulate a concrete situation.

The simulation model is created so that part of the model is generated automatically during the initialization of the simulation model on the basis of the input data. The behaviour of the model is managed by data saved in the MS Access database.

Parts of the simulation model generated on the basis of input data:

- list of resources used,
- list of transport equipment used,
- list of product components,
- shift models for resources,
- capacity of resources,
- capacity of workplaces,
- animation of resource utilization,
- animation of workplace utilization.

Model behaviour managed by input data:

- arrival time for component to production,
- technological information for production:
  - operation time,
  - transport time,
  - resources used,
  - transport equipment used,
  - workplace for technological step,
  - storage position of finished product for assembly (storage workplace or manufacturing workplace),
  - request for assembly,
  - transport of all requested components for assembly to target workplace
  - finalizing production of whole product,
  - etc.

The variable part of the model is generated automatically during the initialization of the simulation model. Visual Basic Application (VBA) is used to generate this. Data for generation is loaded from MS ACCESS database. Data is automatically transferred to modules used in the ARENA simulation system.

2.8. Simulation experiment

Simulation run is started after the variable part of the model has been automatically generated. An animation of the part manufacturing workshops can be seen in the following figure.

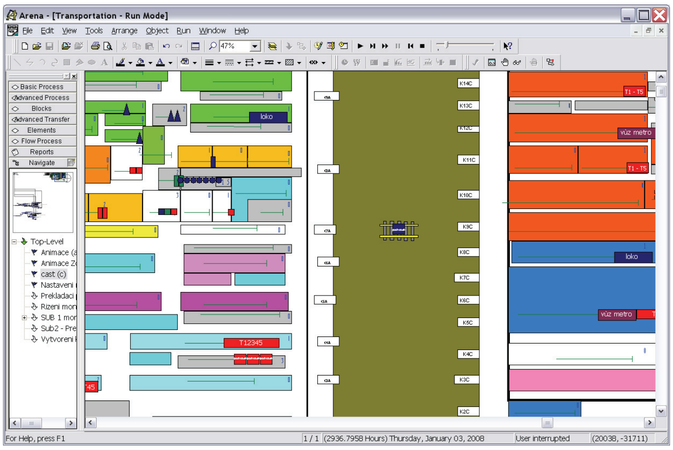


Figure 3 Animation of manufacturing workshops

Animation of workplace in detail is shown in Figure 5. During animation the following information is shown:

- current number of components,
- name of workplace,
- icon of component.

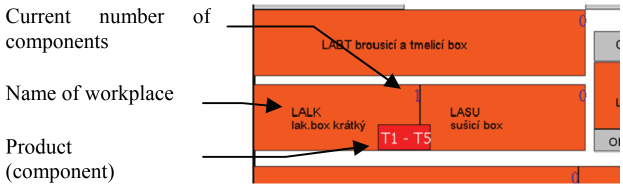


Figure 4 Workplace animation

2.9. Transformed results

After each simulation run an evaluation of the results must be performed. All results are automatically converted into MS Excel tables and graphs and MS Project.

Types of result are:

- production process,
- statistics,
- resources,
- products.

On the basis of the results it is necessary to experimentally optimize the system. Changes for further experiments are saved to modified Excel files. Further steps are the same as in the first experiment (Figure 1).

2.9.1. Production process

The following outputs are included in this group of statistics:

- start time and end time of each operation according to outputs from simulation model
- start time and end time of each hierarchical component of product (Figure 6)

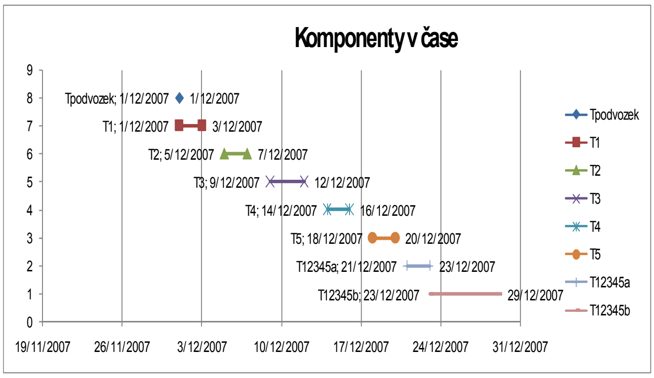


Figure 5 Start time and end time of each hierarchical component

2.9.2. Statistics

In this group of outputs are included simulation outputs, which describe the simulated system from a more global point of view.

We can find the following outputs:

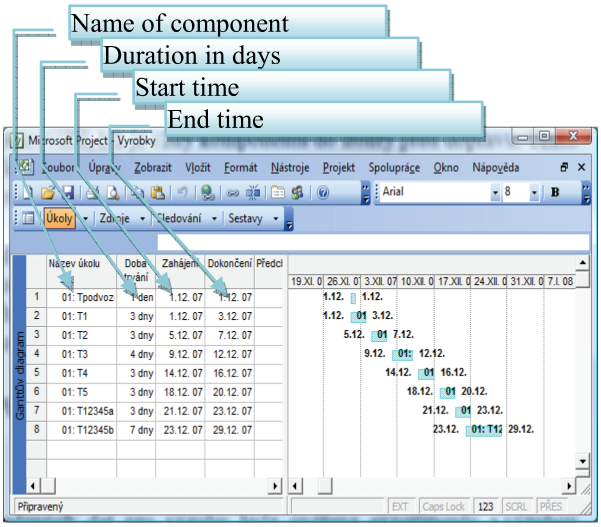
- statistics about utilization of individual resources,
- the number of products waiting in various queues at workplaces,
- etc.

2.9.3. Resources

This group of outputs shows a detailed view of each resource usage over time. In this group of statistics we can see resource usage over time.

2.9.4. Products

This file contains information about the components of each product. Information is about the start time and end time of each hierarchical component and each product. An example of output is shown in Figure 7.



	Název úkolu	Doba trvání	Zahájení	Dokončení	Předch
1	01: Tpodvoz	1 den	1.12. 07	1.12. 07	
2	01: T1	3 dny	1.12. 07	3.12. 07	
3	01: T2	3 dny	5.12. 07	7.12. 07	
4	01: T3	4 dny	9.12. 07	12.12. 07	
5	01: T4	3 dny	14.12. 07	16.12. 07	
6	01: T5	3 dny	18.12. 07	20.12. 07	
7	01: T12345a	3 dny	21.12. 07	23.12. 07	
8	01: T12345b	7 dny	23.12. 07	29.12. 07	

Figure 6 Outputs about time of components

3. OPTIMIZATION OF SYSTEM

During the verification of the whole system, the following optimization of manufacturing system was also performed.

In Figure 8 we can see how much selected storage space was used in one workplace during 3 experiments (v5, v6, v7). From the figure it is evident how organization and management decisions step by step decrease the demand for storage space in the selected workplace.

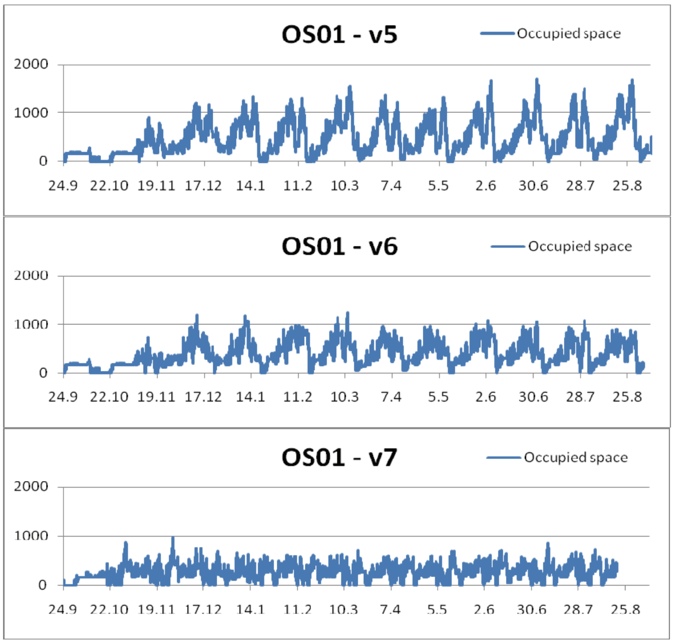
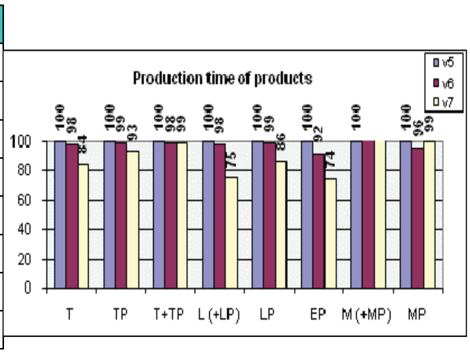


Figure 7 Occupation of selected storage place in experiments

In Figure 9 shows how total times of separate, basic types of products are shortened in individual experiments. For some products we can see that the total time is about 26% shorter between experiments.

	v5	v6	v7
T	100	98	84
TP	100	99	93
T+TP	100	98	99
L (+LP)	100	98	75
LP	100	99	86
EP	100	92	74
M (+MP)	100	101	101
MP	100	96	99



v5	basic experiment
v6	increasing number of grinding workplaces
v7	changing of arrival time for component to production

Figure 8 Total time of products

4. CONCLUSION

The simulation project verifies the group of variants for various capacity utilizations of a production hall and for various production plans. The experiment located bottlenecks, which were eliminated in various ways:

- Some workplaces were moved
- Capacity in some workplaces was increased
- Some handling space and storage space was increased
- Some technological steps were changed

This simulation project enabled many potential problems to be dealt with before real production started. In the future a simulation model will enable verification of throughput of production for various production outputs and for a defined combination of products.

Outputs from this project are used to assist in TOP management decision-making in the company ŠKODA TRANSPORTATION.

REFERENCES

Banks, J., Carson, J.S., Nelson, B.L., Nicol, B.N., 2001. *Discrete-Event System Simulation*. Upper Saddle River: Prentice Hall.

Ulrych, Z., 2006. Generation of simulation models based on data from EIS. *MITIP 2006*. pp. 397-408. September 11-12, Budapest (Hungary).

Bradlay A., 2006., *ARENA documentation*. Rockwell Software Inc.

AUTHOR’S BIOGRAPHY

Vaclav Votava had worked as a researcher at SKODA company in Pilsen (since 1964 in the field of the automatic control, measurements, information systems, software development). From 1991 he has been Associate Professor in the Department of Industrial Engineering and Management at the University of West Bohemia in Plzen. Current activities are focused on education (present and on-line form), simulation of the discrete systems, information systems analysis, multimedia approach, internet information systems.

Zdenek Ulrych works as a lecturer and research worker in the Department of Industrial Engineering and Management at the University of West Bohemia in Plzen (Czech Republic). He holds M.Sc. and Ph.D. in Mechanical Engineering at the same university. His research interests are oriented towards the design and development of eLearning support software and discrete simulation.

Milan Edl works like senior lecturer and research worker at the Department of Industrial Engineering and Management at the University of West Bohemia in Plzen (Czech Republic). He holds M.Sc. and Ph.D. in Mechanical Engineering at the same university. His research interests are oriented to the design and development of multimedia educational software and distributed simulation and their visualization.

Michal Korecky is the Vicechairman of the Board and Development Director of the company SKODA HOLDING a.s. and Chief Operating Officer and Executive of its daughter company SKODA

TRANSPORTATION s.r.o. in Pilsen. He holds M.Sc. in Foreign Trade at the University of Economics in Prague and Ph.D. in Industrial Engineering and Management at the University of West Bohemia. His professional interests are oriented towards business strategy, R&D management, production management and project risk management.

Vaclav Trkovsky works in Development Department of SKODA HOLDING a.s., he is responsible for evaluation of development and investment projects and manages particular projects. He holds M.Sc. in Electronic Computers at the University of West Bohemia and Ph.D. in Technical Cybernetics at the Czech Technical University in Prague. His professional interests are oriented towards project evaluation and management and production process analysis and optimization.

A MODEL FOR ENERGY PREDICTIONS OF A HOTEL ROOM

Adriana Acosta Corzo$^{(a)}$, **Ana Isabel González Santos**$^{(b)}$, **Jesús M. Zamarreño Cosme**$^{(c)}$, **Víctor Álvarez Castelló**$^{(d)}$

$^{(a)}$ Automatic Control and Computation Department, Electrical Faculty, ISPJAE C/ 114 47011 Marianao C. Habana,Cuba
$^{(b)}$ System Engineering and Automatic Department, Faculty of Science,
$^{(c)}$University of Valladolid c/Prado de la Magdalena, s/n 47011 Valladolid, Spain
$^{(d)}$ Hotel Energy Staff Havana City

$^{(a)}$ luism.rodriguez@etecsa.cu, $^{(b)}$ anita@electrica.cujae.edu.cu, $^{(c)}$jesusm@autom.uva.es

ABSTRACT

In this paper a model for energy predictions in three different rooms in a Havana hotel is obtained. The method is based in the determination of the cooling load of the hotel rooms using method Radiant Time Series, RTS. The model was verified with real values of energy consumption in hotel and the results are promissory.

Keywords*:* Energy saving, Modelling, Predictions

1. INTRODUCTION

It is well known that hotel facilities are characterized by being buildings with a permanent use along the year and where their main objective is to ensure comfort and quality for guests in all services during their stay at the facility. (González 1996).

Because of the characteristics of hotel buildings, on hotel rooms and bedrooms, as well as on buildings in general, the cooling load can be predicted without difficulty during the process of designing the air conditioning system. (ASHRAE, 2007). The great diversity in the design, purpose, and use of hotels and motels makes analysis and load studies very important. Load diversification is due to guest rooms' transient occupancy and the diversity associated with support facility operation.

Hence, a climate control system is required, with great flexibility, to meet the conditions of comfort for the occupants during 24 hours. Wide load swings and diversity within and between rooms require a flexible system design for 24-hour comfort. Besides the opening of windows, the only way to provide flexible temperature control is to have individual room components under individual room control that can cool, heat, and ventilate independently of equipment in other rooms.

As stated in the literature, we have identified that the biggest consumers of electricity in a building are air conditioning systems, water pumping and lighting. (Gómez)

Energy use in commercial buildings represents a direct cost to business, while the thermal comfort, visual comfort and indoor air quality of the indoor environment have a substantial bearing on occupants' productivity. It is more than obvious that improved energy efficiency and reduction of energy cost can have beneficial impacts on the competitiveness, the environment, the health and the well being of citizens. (Nikolaou, Kolokotsa and, Stavrakakis).

The demand for energy and fuel consumption of heating, ventilating and air conditioning has a direct impact on the cost of operating a building and an indirect impact on the environment. (ASHRAE, Chapter 32, 2005) .

According to some previous works, the direct impact of the operation of heating, ventilation and air conditioning energy consumption of a building reaches up to 60% of the total energy consumption, thus the importance of paying special attention to this subject. (FIDE 2004; Mohanty 2004; Nikolaou, Kolokotsa and, Stavrakakis; Trott and Welch 2000).

In the analyzed data of consumers of electricity at the hotel, subject of this study, it is considered that the requirements of the hotel air conditioning system are among 61 and 63% of the total electrical energy that is consumed.

Of great interest to researchers and specialists in automatic control has been the application of many control strategies for better management of the air conditioning system, so that a considerable amount of electricity consumption can be saved, without compromising the required comfort.(González and Zamarreño 2005; Ismail 2003).

In the diversity of strategies applied one can identify predictive control, which uses a mathematical model of the process to predict the future evolution of the controlled variable on the prediction horizon. (De Prada 1996).

Obtaining a model for assessing the energy consumption of centralized air conditioning system that cools the hotel rooms is the main objective of this work, as well as the verification of the correct predictions adapted to the conditions of application.

The strategy used in the estimation of energy consumption of the centralized climate system consists of three basic elements: the calculation of the cooling load of the space under consideration, load and energy consumption of secondary equipment, and finally the energy consumption of primary equipment which refers to the large units with their respective chillers.

As a first step in this strategy, we will focus on determining the cooling load or thermal power of the hotel rooms. Various procedures to calculate this magnitude have been reported. Some of the simplest assume that this value depends only on the temperature outside the area as such (González and Zamarreño 2005). Other methods, more detailed, believe that it is a function of other concepts such as: the effects of solar radiation, internal heat gains, stored heat inside the walls, the effect of the wind, the atmosphere of the building and infiltration.

The calculation procedure that is used in this study to determine the cooling load on the hotel rooms, as a basis for analysis of energy consumption in the centralized air conditioning system of the entire installation, is the method of Radiant Time Series, RTS. (ASHRAE, Chapter 30, 2005)

In this work, there have been several studies with different rooms at different times of the year, which have allowed us to verify the feasibility of the application of the method in the estimation of energy consumption. The location of the surveyed rooms corresponds to different geographic areas of the building, so the behaviour of the cooling load in different areas, where the incidence of solar radiation is different in magnitude and time of the day, can be analyzed.

In a previous work, we proposed a model for predicting the cooling load of a room, to be used in the design of a predictive controller that minimizes the energy consumption of a hotel in Havana, as part of a comprehensive strategy for monitoring and control that takes into account the influence of the efficient use of energy carriers in hotel facilities. (Acosta and González 2007).

In this paper, we propose the evaluation of the results of the model previously obtained in the specific conditions of the mentioned hotel, using as a basis of the survey, data of energy consumption and also room occupancy, during 2007.

Results of the model were compared with actual values of daily energy consumption, which are available in a database.

2. MATERIALS AND METHODS

The radiant time series (RTS) method is a simplified method for performing design cooling load calculations that is derived from the heat balance (HB) method. (McQuiston and Spitler 1992). This method was developed to offer a method that is rigorous, yet does not require iterative calculations, and that quantifies each component's contribution to the total cooling load. In addition, it is desirable for the user to be able to inspect and compare the coefficients for different construction and zone types in a way that illustrates their relative effect on the result. These characteristics of the RTS method make it easier to apply engineering judgment during the cooling load calculation process.

The heat transfer can be accomplished through three processes: conduction, convection and radiation. The presence of such processes or a combination of them depends on the characteristics of the surfaces of the elements involved in the transfer of heat. Conduction and radiation processes have inherent delays.

In any particular room, heat sources are disparate and to determine the total heat load it is necessary to take them all into consideration.

The RTSM calculation procedure is illustrated in figure 1. The RTS method is based on the calculation of the gain for each source of heat, and then considers the delays of the conduction and radiation processes .

For this purpose, it uses the so-called radiant and conduction time factors, which are simply ratios of a series of time that distribute heat gains effectively over time. As heat gains are calculated each hour, which is sufficient to track the behaviour of the load, the time series consists of 24 factors and by definition, every radiant or conduction time series must totalize 100%.

The general procedure for calculating cooling load for each load component (lights, people, walls, roofs, windows, appliances, etc.) with RTS is as follows:

1. Calculate 24 h profile of heat gains components for the selected day (for conduction, first account for conduction time delay by applying conduction time series).
2. Split heat gains into radiant and convective parts for radiant and convective fractions.
3. Apply appropriate radiant time series to radiant part of heat gains to account for time delay in conversion to cooling load.
4. Sum convective part of heat gain and delayed radiant part of heat gain to determine cooling load for each hour for each cooling load component.

After calculating cooling loads for each component for each hour, sum those to determine the total cooling load for each hour and select the hour with the peak load for designing of the air-conditioning system. Repeat this process for multiple months to determine the month where the load peak occurs, especially with windows on southern exposures (northern exposure in southern latitudes),which can result in higher peak room cooling loads in winter months than in summer.

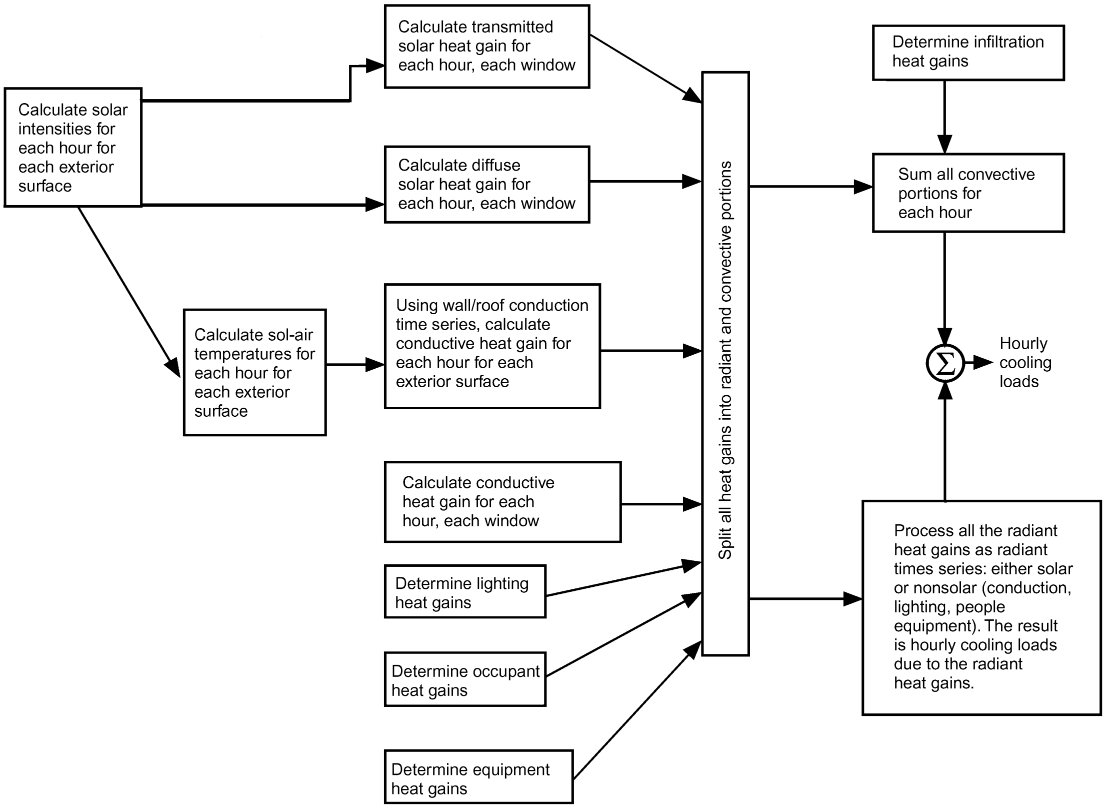

Calculate solar intensities for each hour for each exterior surface
Calculate transmitted solar heat gain for each hour, each window
Calculate diffuse solar heat gain for each hour, each window
Calculate sol-air temperatures for each hour for each exterior surface
Using wall/roof conduction time series, calculate conductive heat gain for each hour for each exterior surface
Calculate conductive heat gain for each hour, each window
Determine lighting heat gains
Determine occupant heat gains
Determine equipment heat gains
Split all heat gains into radiant and convective portions
Determine infiltration heat gains
Sum all convective portions for each hour
Σ
Hourly cooling loads
Process all the radiant heat gains as radiant times series: either solar or nonsolar (conduction, lighting, people equipment). The result is hourly cooling loads due to the radiant heat gains.

Figure 1: Overview of Radiant Time Series Method

In the RTS method, conduction through exterior walls and roofs is calculated using conduction time series (CTS). Wall and roof conductive heat input at the exterior is defined by the familiar conduction equation as:

$$q_{i,\vartheta-n} = UA\,(t_{e,\vartheta-n} - t_{rc}) \tag{1}$$

where
$q_{i,\theta-n}$ = conductive heat input for the surface n hours ago, W
U = overall heat transfer coefficient for the surface, W/(m$_2$·K)
A = surface area, m^2
$t_{e,\theta-n}$ = sol-air temperature n hours ago, °C
t_{rc} = presumed constant room air temperature, °C

Conductive heat gain through walls or roofs can be calculated using conductive heat inputs for the current hours and past 23 h and conduction time series:

$$q_{\theta} = c_0 q_{i,\theta} + c_1 q_{i,\theta-1} + c_2 q_{i,\theta-2} + c_3 q_{i,\theta-3} + \ldots.. + c_{23}\, q_{i,\theta-23} \tag{2}$$

where
q_θ = hourly conductive heat gain for the surface, W
$q_{i,\theta}$ = heat input for the current hour, W
$q_{i,\theta-n}$ = heat input n hours ago, W

The radiant time series method converts the radiant portion of hourly heat gains to hourly cooling loads using radiant time factors, the coefficients of the radiant time series.

$$Q_{r,\vartheta} = r_0 q_{r,\vartheta} + r_1 q_{r,\vartheta-1} + r_2 q_{r,\vartheta-2} + r_3 q_{r,\vartheta-3} + \ldots.. + r_{23} q_{r,\vartheta-23} \tag{3}$$

where
$Q_{r,\theta}$ = radiant cooling load (Qr) for the current hour (θ), W
$q_{r,\theta}$ = radiant heat gain for the current hour, W
$q_{r,\theta-n}$ = radiant heat gain n hours ago, W
r_0, r_1, etc. = radiant time factors

The radiant cooling load for the current hour, which is calculated using RTS and Equation (3), is added to the convective portion to determine the total cooling load for that component for that hour.

3. EXPERIMENTS

The hotel selected for the application of the calculation method is located in the northwest part of the city of Havana, Cuba.

The building has 413 rooms, divided in two blocks. The A-block has 297 rooms in a 9-floor and B-block has 116 rooms in 4 floors. According to its structure and layout, there are three types of rooms: "A", "B" and "C". All measurements were performed in rooms of "A" type, which constitute the majority. For this reason, the building characteristics of the surveyed rooms were similar; the difference comes from their geographical location.

Table 1 shows total rooms by type and geographical location.

Table 1: Total rooms and geographical location

Location	Total rooms	Room Type
N	98 (44 single and 54 double)	A
NW	26	B and C
W	118	A
WS	8	B and C
S	41	A
SE	14	B and C
E	90	A
NE	18	B and C

Inside the room, there is the following equipment: a hair dryer in the bathroom, a coffee machine, a television set, moderate lighting and a small refrigerator. The furniture is abundant. It is believed that the greatest heat gain is the glass door separating the room from the balcony.

Temperature measurements were performed inside the room and on the balcony. We used a four-channel thermometer from "Hanna Instruments". The measurements were performed on February, August and October 2007.

The coefficients of the time series were generated with the help of a program developed by Iu, Pisen (Calvin) of the Faculty of Mechanical and Aerospace Engineering of the State University of Oklahoma, known as “PRF / RTF Generator”.

The program needs as input the constructive detail of the thermal area selecting the types of materials from a database published in the ASHRAE manual (Sowell 1988) .

Table 2 and Figure 2 show the result of the conduction time series generated by the program, for an "A" type room.

The results of the experiments are described in the work (Acosta and González 2007), those compares the behaviour of the load for rooms 7138 and 9109 in A-Block.

In this paper, unlike (Acosta and González 2007), results for three rooms are used: the two mentioned above and a third, 4206, belongs to B-block.

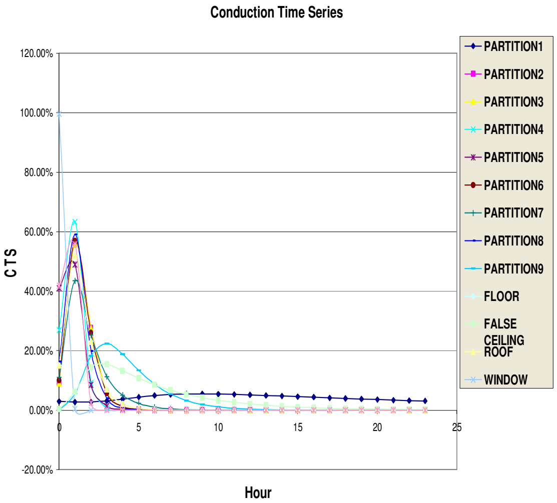


Figure 2: Coefficients of the time series by conduction, CTS, for the walls and divisions of room "Type A".

Table 2: Partition’s Conduction Time Series (CTS)

Surface Name	**PART. 1**	**PART. 2**	**PART. 3**	**PART. 4**
Hour	**CTS**	**CTS**	**CTS**	**CTS**
0	2.90%	9.00%	9.00%	26.92%
1	2.76%	55.76%	55.76%	63.31%
2	2.72%	27.68%	27.68%	9.12%
3	3.05%	6.03%	6.03%	0.62%
4	3.71%	1.22%	1.22%	0.04%
5	4.40%	0.25%	0.25%	0.00%
6	4.93%	0.05%	0.05%	0.00%
7	5.27%	0.01%	0.01%	0.00%
8	5.43%	0.00%	0.00%	0.00%
9	5.46%	0.00%	0.00%	0.00%
10	5.40%	0.00%	0.00%	0.00%
11	5.28%	0.00%	0.00%	0.00%
12	5.12%	0.00%	0.00%	0.00%
13	4.94%	0.00%	0.00%	0.00%
14	4.74%	0.00%	0.00%	0.00%
15	4.53%	0.00%	0.00%	0.00%
16	4.33%	0.00%	0.00%	0.00%
17	4.13%	0.00%	0.00%	0.00%
18	3.93%	0.00%	0.00%	0.00%
19	3.74%	0.00%	0.00%	0.00%
20	3.56%	0.00%	0.00%	0.00%
21	3.38%	0.00%	0.00%	0.00%
22	3.22%	0.00%	0.00%	0.00%
23	3.06%	0.00%	0.00%	0.00%
Sum	100.00%	100.00%	100.00%	100.00%

4. RESULTS AND DISCUSSION

For the implementation of the method in the current application all calculations of the cooling load were programmed on MATLAB language. (MATLAB 2006).

Every day, temperatures outside and inside different rooms were measured. The results are shown in Table 3. It includes the values of the maximum difference between outside and inside temperature of the room, the time of occurrence of such difference and the values of the outside average temperature of the room for each measurement date.

Figure 3 represents the results of measuring the temperature outside for each room. Despite being represented at different seasons, the differences between the average temperatures are not so remarkable.

Table 3: Measured outside temperature

Outside temperature				
Date	Room	ΔTmax [°C]	Hour ΔTmax	Tmean [°C]
23/2/2007	7138	11.4	17:00	24.695
13/8/2007	7138	7.5	12:00	25.937
14/8/2007	7138	9.8	14:00	27.145
18/8/2007	7138	11.1	13:00	28.458
26/2/2007	9109	9.0	15:00	26.004
3/10/2007	4206	7.7	12:00	26.050
4/10/2007	4206	7.1	16:00	26.316
5/10/2007	4206	7.0	14:00	26.341

Nomenclature:

ΔTmax: Maximum difference between outside and inside temperature

Hour ΔTmax: Time of occurrence of the maximum difference

Tmean: Average values of the outside temperature

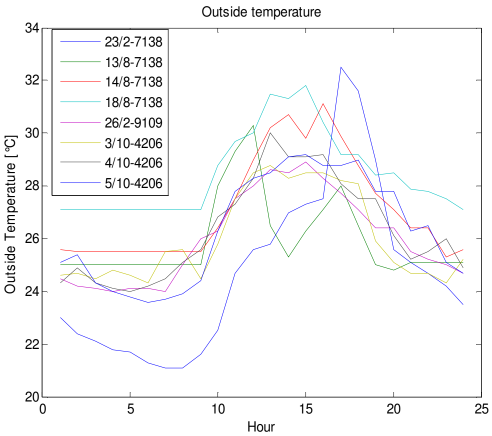


Figure 3: Outside temperature of rooms

Table 4 shows observed values of the cooling load as determined by the implementation of the RTS method, considering the temperature measurements from Figure 3.

Table 4: Cooling load determinate by method RTS

Cooling load by RTS						
Date	Room	qmax [kW]	Hour qmax	qmin [kW]	Hour qmin	qmean [kW]
23/2	7138	2.980	17:00	0.503	6:00	1.140
13/8	7138	3.151	17:00	0.588	6:00	1.259
14/8	7138	3.221	17:00	0.633	6:00	1.332
18/8	7138	3.268	17:00	0.696	6:00	1.394
26/2	9109	3.076	9:00	0.456	6:00	1.139
3/10	4206	1.363	14:00	0.453	6:00	0.791
4/10	4206	1.359	20:00	0.440	6:00	0.747
5/10	4206	1.389	20:00	0.451	6:00	0.781

Nomenclature:

qmax: Maximum cooling load

Hour qmax: Time of occurrence of the maximum cooling load

qmin: Minimum cooling load

Hour qmin: Time of occurrence of the minimum cooling load

qmean: Average load of the room.

The behaviour of the values of the load at each room depends on its geographical location, taking into account that the influence of solar radiation is different. One room is located in the west wing, where the sun shines in the afternoon. Another room in the northern wing, where the incidence of the sun is much lower, and one in the east wing where the sun shines in the morning. Figure 4 shows the values calculated for the total cooling load of the room for each hour during each date of study.

As in the total value of the load all sources of heat gain are important (generated by the occupants, equipment in use, etc..), in Room 4206, where the values of fenestration are small compared to other sources, it is not possible to observe the trend of the behaviour of the load as a function of the incidence of solar radiation, it can be related from Figure 5, where fenestration determined for each measurement date can be seen.

Fenestration is an architectural term that refers to the arrangement, proportion, and design of window, skylight, and door systems within a building. For our purposes, fenestration and fenestration systems will refer to the basic assemblies and components of exterior window, skylight, and door systems within the building.

To verify the results obtained through the RTS method, we used electrical consumption records from the hotel, as well as the number of occupied rooms for the dates on which the measurements were taken. Considering the equipment installed in the building, electricity consumption for cooling corresponds to 63% of the total value; so all the calculations have been performed in the 60 to 66% range of electricity consumption which corresponds to the point of view expressed in the literature, (FIDE 2004; Mohanty 2004; Nikolaou, Kolokotsa and, Stavrakakis; Trott and Welch 2000), and in turn allows us to get a range of variation.

Table 5: Energy and occupancy

Energy and Occupancy				
Date	Energy [kWh]	Occupancy	61 % Energy [kWh]	63 % Energy [kWh]
23/2/2007	17060	335	10406,6	10747,8
26/2/2007	20071	343	12243,3	12644,7
13/8/2007	22432	261	13683,5	14132,2
14/8/2007	23837	300	14540,6	15017,3
18/8/2007	27612	334	16843,3	17395,6
3/10/2007	25437	227	15516,6	16025,3
4/10/2007	25757	305	15711,8	16226,9
5/10/2007	26502	208	16166,2	16696,3

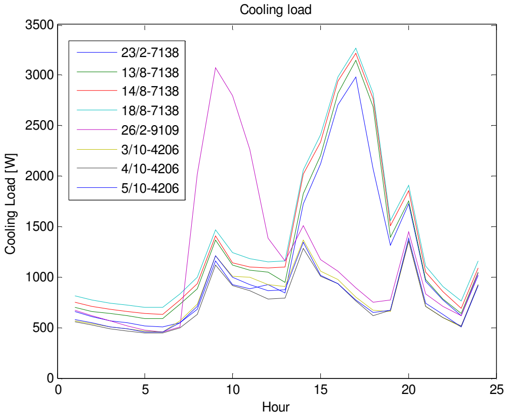


Figure 4: Values of the cooling load calculated for each room.

Table 5 shows the values of total electricity consumption of the hotel, total of occupied rooms and the values corresponding to 61 and 63 % respectively of the total electricity consumption.

During the assessment, it is necessary to consider the nominal electric consumption of the equipment inherent in the climate control system in local offices, restaurants, lounges and other local events in the building, as well as water pumps, fans and cooling units of the fan coils units rooms, so the consumption of only the compressors can be determined. For this purpose, we took into account that the load factor of the engines was 0.75. The nominal power of fan coil units of the rooms was calculated depending on the number of rooms occupied.

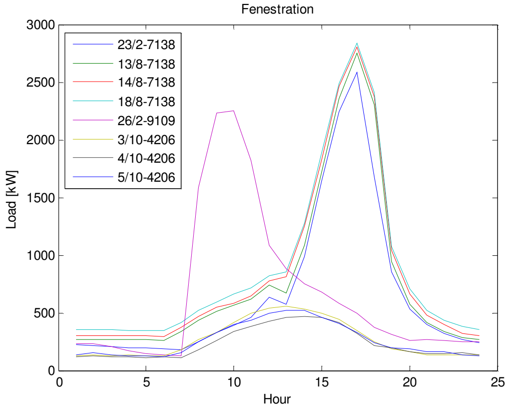


Figure 5: Fenestration values calculated for each room.

The values for the energy consumption of all electric motors, which are involved in the climate control system, taking into account the working hours for each one, was inferred from the total value of electricity consumption hotel (see Table 6) , obtaining electrical energy consumption for the compressors alone. Energy consumption in local offices and restaurants was obtained too by analyzing hotel energy database.

To describe the operation of the compressors, is frequently used, the coefficient of performance known as COP (Coefficient of Performance). This ratio represents the relationship between refrigerating capacity of the compressor and its electricity consumption. (ASHRAE, Chapter 32, 2005). The COP was considered characteristic of the cooling units installed at the hotel, with a value of 2.2.

Table 6: Consumption of compressors

Date	Compressors in "on"	Energy consumption by compressors in "on" [kWh]	Energy consumption by local offices and restaurants [kWh]	Energy consumption by rooms only [kWh]
23/2	11	9675.6	5617.9	4057.7
26/2	11	9675.6	5617.9	4057.7
13/8	12	10555.2	6128.6	4426.6
14/8	12	10555.2	6128.6	4426.6
18/8	8	7036.8	4085.7	2951.1
3/10	13	11434.8	6639.3	4795.5
4/10	13	11434.8	6639.3	4795.5
5/10	13	11434.8	6639.3	4795.5

If we multiply the value of the electricity consumption of a compressor by the corresponding COP, we get the value of the refrigerating capacity. Dividing for 24, we get out the value of the cooling load, which can be removed by the action of the compressors.

Table 7: Cooling load of occupied room calculated taking into account electrical consumption of compressors.

Date	Refrigerating capacity compressors by rooms only [kWh]	Cooling load in total occupied room [kW]	Occup.	Cooling load in each occupied room [kW]
23/2	8926.94	371.95	335	1.110
26/2	8926.94	371.95	343	1.084
13/8	9738.52	405.77	261	1.554
14/8	9738.52	405.77	300	1.352
18/8	6492.42	270.51	334	0.809
3/10	10550.1	439.58	227	1.936
4/10	10550.1	439.58	305	1.441
5/10	10550.1	439.58	208	2.113

Table 7 shows cooling load in each room. These values were obtained by the difference between total energy consumption by compressors in “on” and energy consumption by local offices and restaurants. We get an average value of the cooling load of the hotel rooms dividing the value of the obtained cooling load in total occupied room by occupancy.

Table 8: Comparison of range of cooling load by RTS method and cooling load by real data of compressors consumption

Date	[qmin-qmax] range cooling load by RTS method [kW]	Cooling load by room [kW]
23/2/2007	[0.503 - 2.980]	1.110
26/2/2007	[0.588 - 3.151]	1.084
13/8/2007	[0.633 - 3.221]	1.555
14/8/2007	[0.696 - 3.268]	1.353
18/8/2007	[0.456 - 3.076]	0.809
3/10/2007	[0.453 - 1.363]	1.936
4/10/2007	[0.440 - 1.359]	1.441
5/10/2007	[0.451- 1.389]	2.113

Table 8 presents the intervals of the cooling load obtained by RTS method, and the values of the cooling load, determined taking into account the consumption of the running compressors.

5. CONCLUSIONS

The temperature measurements show the presence of external disturbances affecting their behaviour. Disturbances can be rain, wind speed, clouds and others.

The study confirms what can be found in the literature with regard to the fact that the consumption of air conditioning systems corresponds to a 60 to 66% range of total consumption. For this hotel we found to be between 61 and 63%.

From the results, we conclude that, with the exception of the results corresponding to Room 4206 of Block A, the values obtained for the cooling load for the 7138 and 9109 hotel rooms are in range, comparing them with real data.

Moreover, the values obtained by the application of the method RTS in the 4206 room are lower than those obtained from the actual consumption, which asserts the existence of a reservoir of energy savings, this should be investigated further and will contribute to the success of the implementation of a comprehensive strategy for monitoring and control.

6. FURTHER WORK

It is necessary to perform more measurements, in order to obtain a larger sample of the behaviour of the temperature in different seasons and in different rooms.

It is also necessary to undertake more studies to obtain the power consumption of each individual cooling device.

ACKNOWLEDGMENT

The authors thank the cooperation of the hotel staff, who allowed the realization of the measurements basis of this work and Ms Mercedes Garcia for the English language revision.

The third author thanks the support of the "Ministerio de Educación y Ciencia" through the project "Técnicas avanzadas de supervisión y control para la operación óptima de EDARS".

REFERENCES

Acosta, C.A., González, S.A.I., 2007. Primera aproximación a un modelo de predicciones energéticas de habitación hotelera. *XIII Convención de Ingeniería Eléctrica,* UCLV, Cuba.

ASHRAE, 2005. Chapter 30, Nonresidential cooling and heating load calculation procedures, *ASHRAE Handbook of fundamentals*. Atlanta, GA: American Society of Heating, Refrigerating and Air-Conditioning Engineers, Inc.

ASHRAE, 2005. Chapter 32, Energy Estimating and modeling methods, *ASHRAE Handbook of fundamentals*. Atlanta, GA: American Society of Heating, Refrigerating and Air-Conditioning Engineers, Inc.

ASHRAE, 2007. Chapter 5, Hotels, Motels, and Dormitories, *ASHRAE Handbook of fundamentals*. Atlanta, GA: American Society of Heating, Refrigerating and Air-Conditioning Engineers, Inc.

De Prada, C. 1996. *Fundamentos de control Predictivo de Procesos*, Instrumentación y Control de Procesos: Los manuales de Ingeniería Química.

FIDE, 2004. *Administración de la demanda y experiencias de ahorro de energía eléctrica, Fideicomiso para el ahorro de energía eléctrica*. San Salvador.

González, P.A., Zamarreño, J.M., 2005. Prediction of hourly energy consumption in buildings based on a feedback artificial neural network. *Energy & Buildings* 37/6, pp. 595-601.

González, Santos A. I., 1996. *Sistema de regulación automático para controlar el índice de confort en ambientes climatizados*. Tesis de Maestría. ISPJAE. Cuba.

Gómez, S., *Cuba apunta hacia edificios inteligentes*. XI Convención de Informática, Tribuna de la Habana, edición digital.

IP Seng IU, 2002. *Experimental Validation of the radiant time series method for cooling load calculations*, Oklahoma State University.

Ismail M. Budaiwi, 2003. Air conditioning system operation strategies for intermittent occupancy building in a hot – sumid climate, King Fahd university of petroleum and minerals architectural Engineering Department, *Building simulation.*

MATLAB, 2006. The Language of technical computing. Version 7.2.0.232 (R2006a), The MathWorks, Inc.

McQuiston, F.C. and J.D. Spitler, 1992. *Cooling and heating load calculation manual*, 2nd ed. ASHRAE.

Mohanty B., 2004. Energy efficient air conditioning system, Asia Pro Eco Training Workshop in the Maldives ED72.03 Rational Use of Energy in Industry SMILES MALDIVES: *Building Energy Management*.

Nikolaou, T., Kolokotsa, D., Stavrakakis, D., *Introduction to Intelligent Buildings: INTELLIGENT BUILDINGS: THE GLOBAL FRAMEWORK.*

Sowell, E.F., 1988. Cross-check and modification of the DOE-2 program for calculation of zone weighting factors, *ASHRAE Transactions* 94(2):737-53.

Trott A. R. and Welch T., 2000. *Refrigeration and air-conditioning*, Third edition by Butterworth-Heinemann.

A STOCHASTIC APPROACH TO IMPROVE PERFORMANCE OF PRODUCTION LINES

Cecilia Zanni, Philippe Bouché

LGECO - INSA de Strasbourg
24 Bd de la Victoire, 67084 Strasbourg, France

{cecilia.zanni, philippe.bouche}@insa-strasbourg.fr

ABSTRACT

In our increasingly competitive world, today companies are implementing improvement strategies in every department and, in particular, in their manufacturing systems. This paper discusses the use of a global method based on a knowledge-based approach for the development of a software tool for modelling and analysis of production flows. This method is based on data processing and data mining techniques and will help the acquisition of the meta-knowledge needed for the searching of correlations among different events in the production line. Different kind of techniques will be used: graphic representation of the production, identification of specific behaviour and research of correlations among events on the production line. Most of these techniques are based on statistical and probabilistic analyses. To carry on high level analyses, a stochastic approach will be used to identify specific behaviour with the aim of defining, for example, action plans.

Keywords: Knowledge-based systems, Modeling and simulation of production systems, Discrete event abstraction, Stochastic approach.

1. INTRODUCTION

The search for productivity sources makes the improvement of the contemporary production systems necessary. The production actors are, in a systematic and permanent way, engaged in three stages: the audit, the diagnosis and the search for solutions to improve their production systems.

For the audit of production systems, recent Internet and Intranet technologies allow measuring and storing the state of the different production resources in real time.

From these data, and during the stage of analysis of production flows, the production personnel and the staff in charge must be able to find and formalize the problems inducing a faulty operation of the manufacturing system. Solutions must be imagined in order to increase the productivity at a given cost.

The stages of diagnosis and solution search are, nowadays, primarily instrumented by little formalized expert knowledge. This lack of formalism generates heavy development costs, does not guarantee reproducibility and does not support the necessary knowledge capitalization for the improvement of the production system within the same company. To solve these problems, a solution consists in formalizing the necessary knowledge to set and solve the problems related to that lack of productivity from the data collected during the audit stage. This formalization has to give birth to software tools for assisting the involved actors in a permanent and proactive way.

Several works have been carried out on the performance evaluation of unreliable production lines (Tempelbeier and Burger 2001; Van Bracht 1995; Xie 1993). However, research on the simultaneous consideration of maintenance policies, production planning and quality improvement from an industrial point of view has still to be done.

Confronted with these industrial problems, there are two research lines. On the one hand, there is a great number of scientific works on the detailed modelling of production resources and activities. On the other hand, a much less developed research line is interested in the modelling of problem solving in production systems design. From these two categories, our research group is interested in the understanding and modelling of the field experts reasoning during the stages of production flow analysis and solution searching. We are also interested in automating this reasoning in order to bring proactive software assistance.

This article presents our approach for performance analysis of production flows. The approach is based on statistical and probabilistic methods and will be a new case of application of the stochastic approach (Le Goc, Bouché, Giambiasi 2006). Section 2 presents the industrial context and describes the project. Section 3 presents the data graphical representation before setting the definition of phenomena in Section 4. Section 5 effectively presents the stochastic approch and finally, Section 6 states our conclusions and perspectives of future work.

2. THE INDUSTRIAL CONTEXT

As we have presented in previous communications (Zanni, Barth, Drouard 2007; Bouché, Zanni 2008; Zanni, Bouché 2008), our group is interested in the developing of a software tool for allowing the decision makers companies to have an analysis of their production line flows. This analysis will consist in a general and by-workstation productivity evaluation, the main objective being the maximization of this productivity in terms of the number of produced parts in a given time window.

This diagnosis will be followed by an action plan for the improvement of the line, according to three criteria (quality, maintenance and yield) and a valorisation of the losses which could have been avoided if the action plan was executed. The general idea is to maximize the productivity by improving the production cycle time and by reducing the workstations breakdowns / outages and the number of rejected parts.

We are using a data acquisition system that, after having placed sensors in strategic places of the production line, allows the measuring of different indicators.

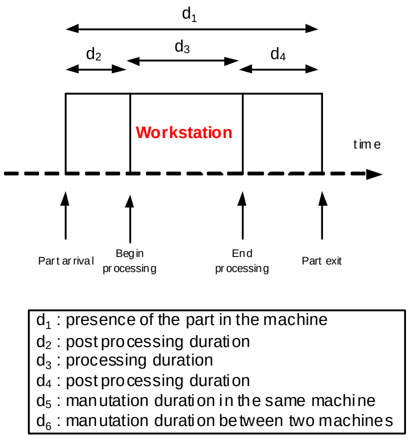

d1
d2
d3
d4
Workstation
time
Part arrival
Begin processing
End processing
Part exit
d1 : presence of the part in the machine
d2 : post processing duration
d3 : processing duration
d4 : post processing duration
d5 : manutation duration in the same machine
d6 : manutation duration between two machines

Figure 1: Indicators to be measured during production

During the production stage, we are able to detect if the part is good or bad (and, eventually, the associated fault code) and the times of (Figure 1):

- The arrival of the part to the workstation,
- The beginning of processing of the part in the workstation
- The end of processing of the part in the workstation, and
- The exit of the part from the workstation.

They aim at defining a set of durations linked with the different stages of the work on the piece on the workstation (Figure 1).

In the case of failure of a workstation, the indicators we measure are:

- The failure beginning time,
- The failure end time,
- The identification code of the failure.

The data acquisition system will also provide other necessary information, in particular, the control parameters of the workstations, i.e. some workstation characteristics which will be specific for each process plan. It will also provide maintenance data, information on production modifications, and other relevant information.

We study production lines such as the one described in Figure 2:

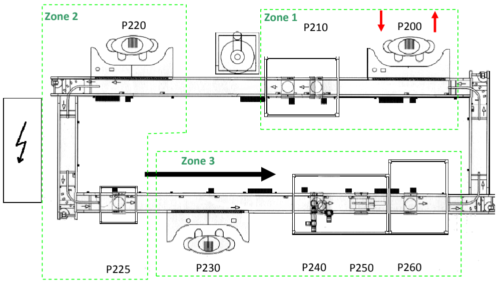


Figure 2: Example of a production line

This is a closed loop, where we have a set of workstations which can be automatic or manual. In Figure 2, workstation P200, for example, is the point where pieces are injected in the line and where they go out. There is also a finite set of pallets that turn around in the loop for transporting the pieces form a workstation to the next one.

This organisation makes necessary to take into account a last set of parameters which are:

- The instance of the production plan,
- The working team.

To take these parameters into account, data are separated by production type and/or by working team; the idea is to guarantee that time periods for analysis are uniform.

3. DATA GRAPHICAL REPRESENTATION

These data can be analyzed with frequencies and sequential methods.

First we can proceed to a Poisson analysis. A Poisson process is a process of enumeration which describes the evolution of a “quantity” in time (Figure 3). In our study, it will be a question of tracing the evolution in time of durations (d_1, d_2,…). In the case of a perfect process, the Poisson curve is a line characterized by its slope λ (λ is a ratio number parts/time).

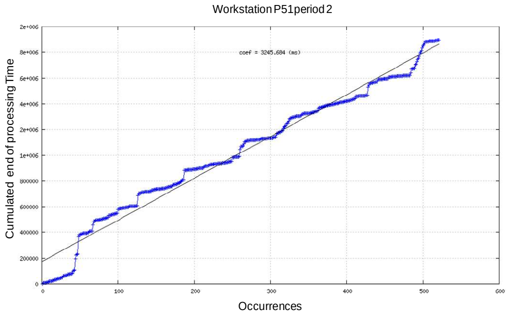


Figure 3 : Poisson process of evacuation times of a workstation

In real processes, we will observe various slopes which will make possible, for example, to determine the moments when the production is “faster”, if there are intervals of drif in the workstation, and to define ranges where the behaviour of the station requires a more thorough analysis.

Second, we can study the evolution of the working time with a model inspirated on control charts which are used in Statistical Process Control (Ishikawa, 1982). We trace the different durations of the tasks according to time (Figure 4).

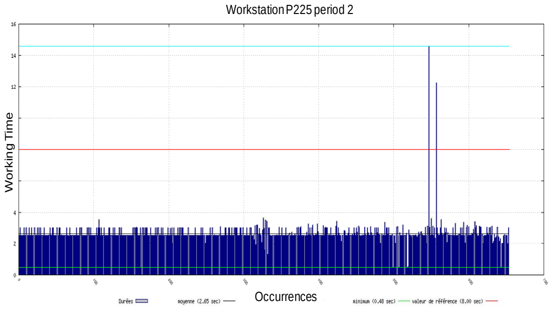


Figure 4: "Control chart" of processing times of a workstation

That will make possible to study possible drifts of the workstation to check that the process is “under control”, to identify the places where improvements could be made, to identify changes of rate/rhythm, or perturbations.

Finally we can analyse properties of the distribution of duration (Figure 5). We trace the frequency of the durations to study the setting under statistical laws of the station to consider.

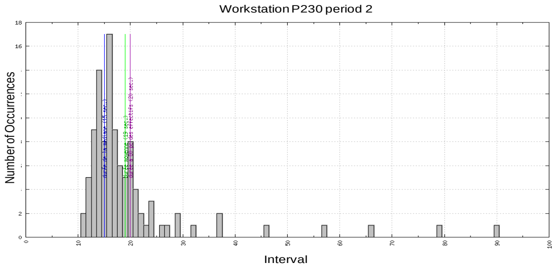


Figure 5: distribution of processing times of a workstation

From these curves, a certain number of analysis may be carried out, such as the analysis of dispersions, of aberrant values, or others.

Other graphical representation could be imagined, such as synthesis representations. But the three types of presented representations are, for us, the base of the analysis. This is a first method to have a better view of reality, and to identify specific zones of bad behaviour or specific phenomena. It corresponds to the more specific level of abstraction. To make better studies, we need to build meta-data which will be associated to specific events or behaviours of the production that might indicate a saturation of stock or a drift of the working time, for example. These behaviours can not be directly deduced from data.

A set of transformations has to be applied the data to obtain the expression of a certain behaviour under the form of a “phenomenon”. The next section will define what we call a phenomenon before showing how we can compute phenomena from our data.

4. PHENOMENA

A phenomenon is the expression of a particular behaviour. A phenomenon will be described by a set of attributes, and at least (Le Goc 2004):

- A name,
- A characterization of the location in the production line,
- Two dates:
  - A begin date
  - An end date

These phenomena have been defined in collaboration with experts of production, and in function of the goal of the analyses. A set of specific phenomena has been retained:

- Stock_Saturation,
- Workstation_Drift,
- Workstation_Instability,
- Not_Adapted_Workstation ,
- Out_Of_Bounds_Time, among others.

These phenomena can be deduced from the data provided by the data acquisition system.

For example if we consider the “Workstation_Drift” phenomenon, it is the expression of a drift of the production time on a workstation. We can observe it in the Poisson curves or the control charts with the study of processing duration (d_3) or the part presence duration (d_1).

More precisely, on the Poisson curves, a phenomenon of drift will be a possible translation of a drop in the “speed” and thus of a drop in the slope of the curve. On control charts, this phenomenon will be the translation of a regular increase in the execution times of a task in time (Figure 6).

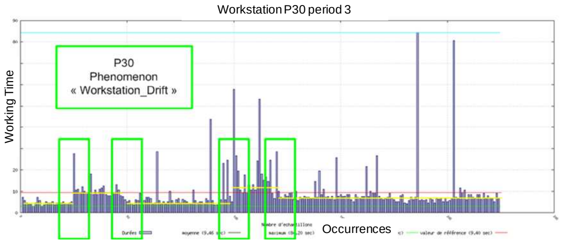


Figure 6: Visual observation of the "Workstation_Drift" phenomenon

To effectively compute occurrences of this phenomenon, we will calculate the slopes of the Poisson processes that are associated with this workstation during the considered time window. If we observe several slopes, named successively λ_1, λ_2, λ_3 ... and that these slopes are decreasing, then we can diagnose a “Workstation_Drift” phenomenon.

$$\forall \lambda_i, \mathrm{i} \in [1, \mathrm{n}], \mathrm{IF}\, \lambda_i > \lambda_{i+1}$$
$$\text{THEN "Workstation\_Drift"}$$

In the same way, the list of the durations of processing on a workstation can be used to diagnose a “Workstation_Drift” phenomenon. It will be characterized by an increase of the average of durations.

$$\forall \mathrm{d}_i, \mathrm{i} \in [1, \mathrm{n}], \mathrm{IF\ Average}(\mathrm{d}_1, ..., \mathrm{d}_k) < \mathrm{Average}(\mathrm{d}_{k+1}, ..., \mathrm{d}_n)$$
$$\text{THEN "Worstation\_Drift"}$$

Let us consider the data in Figure 7:

occurrence	date	added up durations	slope	lambdas
1084	19/09/2007 19:01	18759881		
			97771,96	0,00010233
1388	19/09/2007 20:00	21730557		
			12599,15	0,00007937
1611	19/09/2007 20:53	14540168		

Figure 7: Data of a post of the production line

The slopes values on this manual workstation lead us to say that we are in presence of a “Workstation_Drift” phenomenon from 20h00 to 20h53.

The idea is to calculate the slopes on a temporal horizon that is coherent with the production line speed and the considered station and to compare the slopes regularly. The algorithm to build occurrences of this phenomenon is depicted in Figure 8:

```
Constant : h time windows characterized by two dates $d_1$ (begin) et $d_2$ (end)
Constant : p tolerance
Variables : $\lambda_1$ past slope, $\lambda_2$ curent slope
Boolean : C Boolean uses to know if we are in Workstation drift period (1) or not (0)

Data :      NbOc ($d_i$) Number of pieces treated at time $d_i$
                DC ($d_i$) Cumulated period of work at time $d_i$

Temporal loop on h
        Compute $\lambda_2$ = (NbOc($d_2$) – NbOc ($d_1$))/(DC($d_2$)-DC($d_1$))
        Compare $\lambda_1$ et $\lambda_2$
        IF the difference is upper than p
            THEN IF  C = 0
                THEN    Init occurrence of phenomenon Workstation_Drift
                        Begin date phenomenon = $d_1$
                        C=1
        IF difference is lower than p
                THEN IF C = 1
                        THEN    Stop occurrence of phenomenon Worstation_Drift
                                End date phenomenon = $d_1$
                                C=0
        Make change the time, actualize $d_1$ et $d_2$
        Change value $\lambda_1$ with value $\lambda_2$
```

Figure 8: Algorithm for identifying the "Workstation_Drift" phenomenon

The application of this on data of Figure 6 will give us 4 occurrences of the “Workstation_Drift” phenomenon.

To have another example, let us consider the “Stock_Saturation” phenomenon. It can be deduced from data of post processing durations (d_4). Figure 9 shows an example of a time window for workstation P210 where we can observe 5 occurrences of this phenomenon.

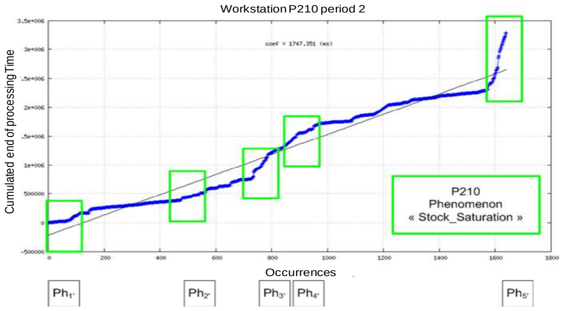


Figure 9: Example of the phenomena "Stock_Saturation"

It is important to remark that now we have to consider 5 occurrences of the “Stock_Sturation” phenomenon rather than all the data on the same time window. We make, also, the assumption that, if no phenomenon is detected, the behaviour of the production line is correct.

Therefore, because phenomena are meta-knowledge, we are able to build a sequence of phenomena, which contains more information than the original data but “lighter” than the original set.

The construction of phenomena is then a kind of discrete event abstraction (Le Goc 2004).

Post analysis may be performed on the phenomena sequence, by application of the stochastic approach to identify correlations which can exist between phenomena. Next subsection will show the bases of the stochastic approach and how we can use it to obtain fault models. These models will serve in the last step of our development to built action plans.

5. THE STOCHASTIC APPROACH: IDENTIFICATION OF FAULT MODELS

Once the log of phenomena obtained, new studies may be carried out.

New frequencies studies of the same type as described before may be made, but more especially we can carry out probabilistic studies, in order to identify if there exist correlations among phenomena, and the temporal constraints on these correlations.

The objective is to produce “fault models”. These models may be used to perform real time diagnosis, but also to define action plans and corrections on the production line. Fault models will probably reveal implicit links among the workstations of the considered line.

With this objective in mind, we will apply the stochastic approach (Le Goc, Bouché, Giambiasi 2006). This approach will permit the identification of sequential relations which can exist among phenomena and the computation of time constraints to label those sequential relations.

The stochastic approach is based on the representation of a sequence of discrete event classes in the dual forms of a homogeneous Markov chain and a superposition of Poisson processes.

We will use the BJT4S algorithm (BJT4S is the acronym of “Best Jump with Timed constraints For Signature”). The role of this algorithm is to identify the most important sequential relations from the Markov chain model and to compute timed constraints on these relations from the Poisson process model. The union of sequential relations and timed constraints constitutes a “signature”. It is an operational model of chronicle which its anticipating ratio is equal to or higher than 50% (the anticipating ratio is a measure of the quality of the model). A signature is a behavioural model representative of certain specific situations.

The application of the stochastic approach corresponds to the generic level of analysis of our project. The stochastic approach produces behavioural models that, according to our experience, are realistic indeed and can be used to make prediction or diagnosis for example.

Figure 10 shows an example of correlation between two phenomena detected on workstations P210 and P220 (See the production line in Figure 2). We detect the “Out_Of_Bounds_Time” phenomenon on workstation P220 and the “Stock_Saturation” phenomenon on workstation P210. It is easy to see that, if there is an important working time on workstation P220, the line will be slowed down and a consequence is the saturation of stock that can be observed on workstation P210.

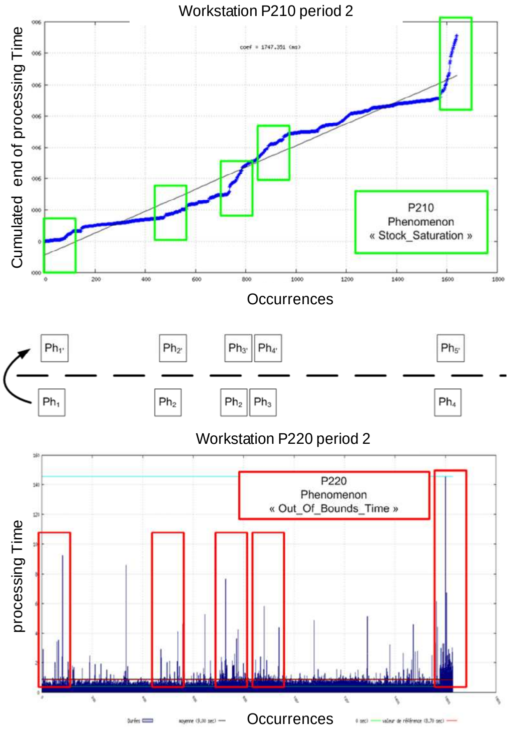


Figure 10: example of correlation between phenomena

Therefore, to improve performance of the line in this context, the problem will not be to eliminate the “Stock_Saturation” phenomenon but to improve the working time on workstation P220. With a single action we can act on two phenomena.

In our context, we make the assumption that the stochastic approach will permit to define global models like the one in Figure 11:

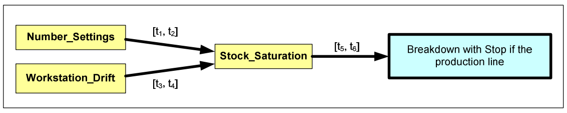


Figure 11 : Example of Breakdown Model

This model has to be read in the following way: If we observe an increase in the number of machine settings followed by a saturation of the stock in the time interval [t_1, t_2], or if we observe a drift in the behaviour of the workstation followed by a saturation of the stock in the time interval [t_3, t_4] then we have the risk to have a breakdown with the stop of the production line in the interval [t_5, t_6].

These models will be the base to the proposal of action plans to improve the performance of the production line in study.

They can also be used to make new studies, define new phenomena with high abstraction levels, etc. This phase will be described in more detail once the implementation of the algorithms for building phenomena will be achieved.

6. CONCLUSION

We have presented our global method based on a knowledge-based approach for the development of a software tool for modelling and analysis of production flows.

To the best of these authors understanding, the reasoning on the number of produced parts and the recommendations according to the three criteria, quality, maintenance and yield, have not been fully addressed yet. Also, the generic vs. specific analysis approach will make the tool flexible and available for use by the production staff on site (not necessarily at ease with other possible performance indicators) and decision makers.

The method we propose is based on data processing and data mining techniques. Different kind of techniques will be used: graphic representation of the production, identification of specific behaviours to identify phenomena, and research of correlations among them on the production line. Most of these techniques are based on statistical and probabilistic analyses. To carry on high level analyses, a stochastic approach is used to identify breakdown models.

Breakdown models can finally be used to propose action plans, which can be studied by simulation before implementation.

Therefore, last step of our project will be the development of a simulator. We will use it to compute the effects of the action plan. The principle will be to build new sequences of data with the specifications of the action plan and to introduce them in real data to compute the effects. If a proposition of an action has no effect, it is not necessary to implement it. Furthermore, if the implementation of that action does not produce significant improvements (according to the decision maker) in the quantity of good pieces that were produced, a non application of the action might be envisaged.

REFERENCES

Bouché, P., Zanni, C., 2008. A Stochastic Approach for Performance Analysis of Production Flows. *Proceedings of ICEIS 2008, 10th International Conference on Enterprise Information Systems*. 12-15 June, Barcelona, Spain.

Ishikawa, K., 1982. *Guide to Quality Control*. Unipub / Quality Resources.

Le Goc, M., 2004. The discrete event concept as a paradigm for the perception based diagnosis of sachem. *Journal of Intelligent Systems* 8(3/4), 239–290.

Le Goc, M., 2004. SACHEM, a Real Time Intelligent Diagnosis System based on the Discrete Event Paradigm. Simulation. *The Society for Modeling and Simulation International Ed.*, vol. 80, n° 11, pp. 591-617.

Le Goc, M., Bouché, P., Giambiasi, N., 2006. Temporal Abstraction of Timed Alarm Sequences for Diagnosis. *Proceedings of COGIS'06, COGnitive systems with Interactive Sensors*. Paris, France.

Tempelbeier, H., Burger, M., 2001. Performance evaluation of unbalanced flow lines with general distributed processing times, failures and imperfect production. *IEE Transactions* 33(4), 419–446.

Van Bracht, E., 1995. Performance analysis of a serial production line with machine breakdowns. *Proceedings of IEEE symposium on emerging technologies and factories automation*. Paris, France.

Xie, X., 1993. Performance analysis of a transfer line with unreliable machines and finite buffers. *IEE Transactions* 25(1), 99–108.

Zanni, C., Barth, M., Drouard, L., 2007. A Knowledge-Based Tool for Performance Analysis of Production Flows. *Proceedings of IFAC MCPL 2007 – The 4th International Federation of Automatic Control Conference on Management and Control of Production and Logistics*. Sibiu, Rumania.

Zanni, C., Bouché, P., 2008. A Global Method for Modelling and Performance Analysis of Production Flows. *Proceedings of EUROSIM/UKSIM 2008 10th International Conference on Computer Modelling and Simulation*, p740-745. 16-19 June, Emmanuel College, Cambridge, England.

CACSD TOOL FOR SIMULATION AND PERFORMANCE OF MULTI-RATE SAMPLED-DATA SYSTEMS

Yolanda Cerezo[a], Ignacio López[b], Alfredo Cuesta [c], Luis Grau[d]

[a] Universidad Francisco de Vitoria. Madrid. Spain.
[b,d] Universidad Nacional de Educación a Distancia. Spain.
[c] Felipe II College, Universidad Complutense de Madrid. Spain

[a] y.cerezo.prof@ufv.es, [b] ilopez@scc.uned.es, [c] acuesta@cesfelipesegundo.com , [d] lgrau@scc.uned.es

ABSTRACT

In this work a CACSD tool named MRPIDLAB is presented. The tool is devoted to the analysis and tuning of PID controllers working on a general multirate sampled data system, where different variables are sampled at different rates. The tool allows testing different scenarios including not only time constrains in the sampling rate but also delays in every channel. Also, since a multi-rate PID (MRPID) regulator, with 3 extra degrees of freedom which are the internal action rates, is obtained a new general tuning method is implemented. Such a method is a multi-objective optimisation in which up to four performance requirements might be imposed.

An illustrative example is given to show the utilization of the developed analysis tool and to demonstrate the usefulness of the proposed multirate PID control and its tune.

Keywords: Multi-Rate control, Digital control, PID tuning, CACSD tools

1. INTRODUCTION

Multirate (MR) Digital Control Systems are those in which more than one variable is updated at different rate (Velez and Salt 2000). Multirate Digital Control is a research field of great interest, since it can be applied to many different practical situations, both to improve the system performance and to deal with situations which are inherently multirate (Cuenca 2004). The two main fields of application are motion control, from the seminal works of Araki to the most recent ones by Gu Tomizuka or Chen (Gu and Tomizuka 2000; Tomizuka 2004; Mizumoto 2007), and network based control either on a field-bus or even on internet (Sala 2005; Casanova et al. 2006; Salt et al. 2006; Yang and Yang 2007).

On the other hand, PID controller is still the most widely used in industrial processes (Aström and Hägglund 2000; González, López, Morilla and Pastor 2003). In general, its parameters are tuned to achieve a requested evolution in the controlled variable for changes either in the reference or in the load (Aström and Hägglund 2005). In a well designed control system, features of both sensors and actuators should agree with the dynamics of the system to be controlled. Despite of that, it could happen that the measure is not always available at one particular instant specified by the controller, the actuator input can not or should not be modified in certain circumstances or the evolution of the control signal is not adequate for the actuator. Moreover, the controller also imposes its own constraints. However, using a discrete controller has the advantages of obtaining the same responses of a continuous one if the sampling is fast enough but also provides new control strategies impossible to achieve in the continuous domain (López, Dormido and Morilla 1994).

Since the controller is the element that makes the system output to follow the reference satisfying different specifications at a time, it would be desirable not to impose an excessive number of constraints but still being versatile enough to adapt well to the constraints imposed by other elements in the system. To this end, this work extends to the multirate case the classic structure of a sampled control system with PID controller. It also describes the structure of a multirate sampled-data control system, which uses a discrete multirate PID controller. In this system, different periods are used to take samples from the process output, reconstruct the control signal and calculate the control actions, with the only constrain of these being constant but not necessarily equal one to another. The performance of the system is described from a discrete point of view.

A CACSD tool developed for Matlab is presented. Previous works in CACSD tools for MR control systems can be found in (Velez and Salt 2000; Albertos et al. 2003; Cuesta, Grau and Lopez 2006). The contributions of the tool here presented are the implementation of the MRPID controller, in its different configurations (interactive and non interactive form), via a graphical user interface (GUI). The use of an optimization algorithm, particularly using heuristic methods, constitutes a global tool to the design and tuning of multirate PID controllers for a wide range of control engineering applications.

On the other hand the tool is flexible enough to allow testing the effect on the performance of different factors like load disturbances, noise measurements, saturation in the actuators not only in the MR case but also in Single-Rate and even continuous systems

This paper is organised as follows. The next section describes the control systems and the different working modes than can be set. In section III the tool is presented. The tuning method is proposed in section IV. An example is given in section V in order to clarify the use of tool and the benefits of MR control in some cases. Finally the conclusions are given.

2. DESCRIPTION OF THE CONTROL SYSTEM

Fig. 1 shows the structure of the control system. It is a typical sampled control loop (Franklin, Powell and Workman 1994), where both load perturbations d(t) and a noise in the measure n(t) have been included. Note that period T_y, used for taking samples of the control variable and period T_u, used to modify the controlled variable can be different. In the sequel, this system will be called *multirate* system when $T_y \neq T_u$, or *single-rate* system when $T_y = T_u$. Note that the latter is only a particular case of the former.

Following with the same figure, some other signals are present: variable to control y(t), discrete signal of the variable to control $y^* = y(k_y T_y)$ (obtained at period T_y), discrete reference signal $r^* = r(k_y T_y)$, discrete error signal $e^* = e(k_y T_y) = r^* - y^*$, discrete control signal $u^* = u(k_u T_u)$ (obtained at period T_u), continuous control signal u(t) (obtained from u* using a holder H at period T_u).

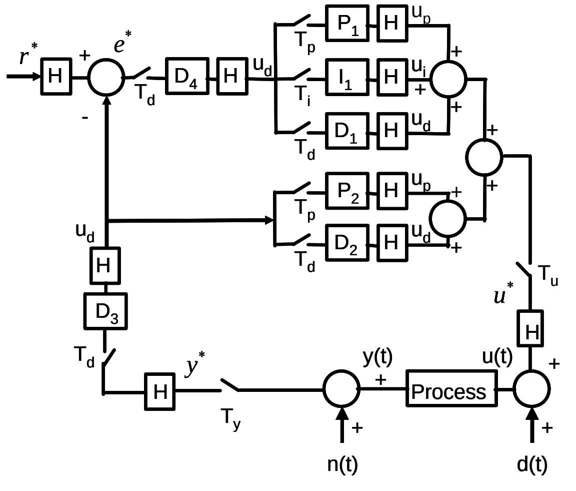


Figure 1: General structure of the multirate PID into a control system.

Taking the classical multirate structure as a starting point, this work proposes its generalization through the introduction of new structures for the controller, multirate PID, in which the proportional P, integral I and derivative D control actions are obtained with periods T_p, T_i and T_d, respectively (which are allowed to be different and independent to each other). H blocks that also appear in this figure are holders that allow the generation of signals of period whilst the last value of its input remains constant between successive samples. T_h can be chosen as the minimum significant number among all periods of the global system. This allows considering any value for the periods with such precision. Note that the digital sum can be obtained, since all periods are multiples of T_h. In what follows, this controller will be called *Multirate PID Controller*. The input of the controller is the error signal e*, which is held at period T_h to generate signal $e(k_h T_h)$. Each of these actions are calculated at those periods and outputs $u_p(k_p T_p)$, $u_i(k_i T_i)$ and $u_d(k_d T_d)$ are obtained and, again, held in order to obtain the sum of all of them, obtain $u(k_h T_h)$ and subsequently sample at period T_u to get the controller output u*.

From the general structure presented and based on different relationships between periods (as proposed in the structure) some operating modes can be defined:

Mode 1. Cuasi-continuous: $T = T_r = T_y = T_u = T_p = T_i = T_d$, with T small enough to design the controller in the continuous domain. This technique is known as Digital Redesign and it is widely applied.

Mode 2. Single-rate: $T = T_r = T_y = T_u = T_p = T_i = T_d$, with T greater than in Mode 1 so that a direct digital design must be chosen.

Mode 3. Single-rate process with multirate controller: $T = T_r = T_y = T_u = m_p T_p = m_i T_i = m_d T_d$.

Mode 4. Multirate process with single-rate controller: $T_r = T_y \neq T_u$, $T_p = T_i = T_d$.

Mode 5. Multirate process with multirate controller: $T_r = T_y \neq T_u$, T_p, T_i, T_d such that at least one of them is different from the rest.

A general framework is now considered in order to deal with different multirate controllers, such as PID, PI-D, I-PD, both interactive and non interactive. Based on each type of controller, only the three corresponding actions will be considered:

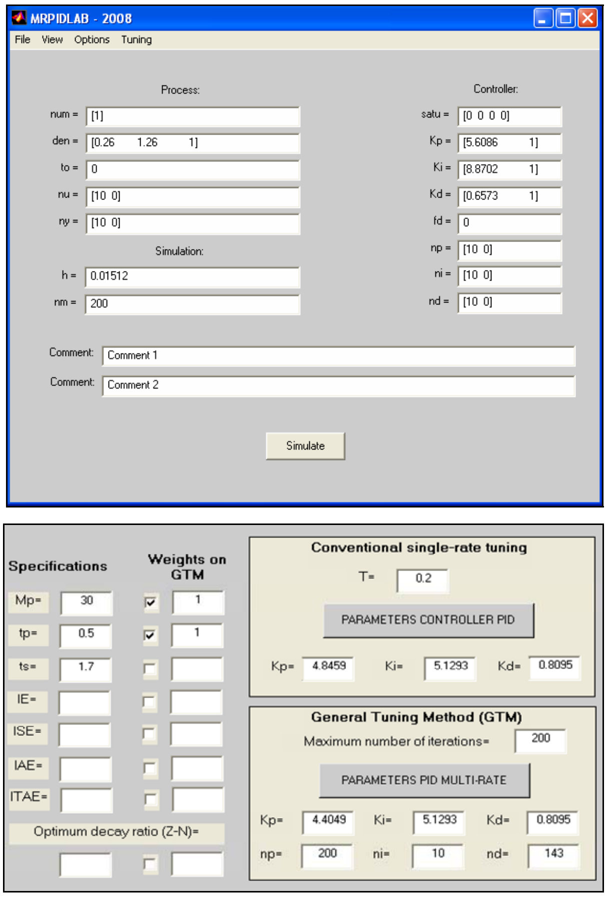
1. non interactive PID, with P_1, I_1, D_1.
2. non interactive PI-D, with P_1, I_1, D_2.
3. non interactive I-PD, with P_2, I_1, D_2.
4. interactive PID, with P_1, I_1, D_4.
5. interactive PI-D, with P_1, I_1, D_3.
6. non interactive I-PD, with P_2, I_1, D_3.

The model of the MRPID, of the process and of the control loop are then implemented as algorithms into the MRPIDLAB. Although there are modelling techniques that provide theoretical results for MRPIDs (Salt and Albertos 2005; Cuesta, Grau, López, 2007) they impose simple ratios between the error rate and the control rate. In this sense the numerical model proposed here is better than these because it allows any ratio.

3. SIMULATION TOOL

Both the simulations presented in this work and the tunings have been obtained using a CACSD tool named

MRPIDLAB and developed by the authors for Matlab. It consists of two interfaces shown in Fig. 2: The input interface (above) and the General Tuning Method (GTM) (below).


MRPIDLAB - 2008
File
View
Options
Tuning
Process:
Controller:
Simulation:
Simulate
Specifications
Weights on GTM
Conventional single-rate tuning
PARAMETERS CONTROLLER PID
General Tuning Method (GTM)
PARAMETERS PID MULTI-RATE

num = [1]
den = [0.26 1.26 1]
to = 0
nu = [10 0]
ny = [10 0]
h = 0.01512
nm = 200
satu = [0 0 0 0]
Kp = [5.6086 1]
Ki = [8.8702 1]
Kd = [0.6573 1]
fd = 0
np = [10 0]
ni = [10 0]
nd = [10 0]
Comment: Comment 1
Comment: Comment 2

Mp= 30 ☑ 1
tp= 0.5 ☑ 1
ts= 1.7 ☐
IE= ☐
ISE= ☐
IAE= ☐
ITAE= ☐
Optimum decay ratio (Z-N)= ☐
T= 0.2
Kp= 4.8459 Ki= 5.1293 Kd= 0.8095
Maximum number of iterations= 200
Kp= 4.4049 Ki= 5.1293 Kd= 0.8095
np= 200 ni= 10 nd= 143

Figure 2: Input interface (above) and the GTM interface (below)

3.1. Input interface

Here, the controls are distributed in four sections: Simulation, Process, Controller and Comments.

- *Simulation*, which allows to introduce the simulation period h (T_h or one of its integer submultiples) and the number of samples to calculate nm. T_h can be chosen as the minimum significant number among all periods of the global system. This allows considering any value for the periods with such precision.
- *Process*, which allows to introduce the process model (linear system with time delay, *num*, *den* and t_0), periods $T_y=n_y h$, $T_u=n_u h$ and delays associated to y* and u*.
- *Controller*, which allows to introduce saturations for the control signal u(t) and the output y(t), as well as control parameters (K_p, K_i, K_d), the algorithm applicable to each action, the factor of the derivative filter f_d, periods $T_p=n_p h$, $T_i=n_i h$, $T_d=n_d h$, and delays associated to the different actions.
- *Comments*, lines for comments associated to the simulation.

In the lower part of the window a 'Simulate' button runs a simulation with the data introduced.

There are also four different menu options:

- *File (open, save* and *exit)*, which allows to save in a file all variables associated to an experience, or open a file of such type, and exit the application.
- *View (all, output, error, control, proportional, integral, derivative, measure, actuator* and *delete)*, which allows to see the different signals obtained through the simulation and to delete all but the last simulation.
- *Options (controller-type, load, noise* and *anti-windup)*, allows to among controllers PID, PI-D or I-PD, interactive or not, and introduce perturbations in load, noise in measure and anti-windup methods.
- *Tuning (frequency and time)*. Depending on the specifications, it allows to do a pre-tuning and a fine fit for the final tuning, by means of numerical optimization and using heuristics algorithm.

MRPIDLAB proves to be a powerful tool especially in the controller design stage because it provides an environment in which different uncertainties and time scenarios, especially multi-rate situations, can be included, obtaining more realistic simulations.

3.2. GTM Interface

GTM interface is launched when tuning is requested. Up to four different specifications can be set: Maximum percentage overshoot (M_p) and the time it occurs (t_p), settling time (t_s) and one of the performance criteria among IE, ISE, IAE or ITAE. Alternatively also the decay ratio can be set. To its right the weights for each specification must be fixed.

The upper box to the right of the interface is necessary to find the parameters of a single-rate PID controller that satisfies or gets the best control action given the specifications. This is the starting point for the optimisation (box below-right) for tuning the real multi-rate controller, implemented in the tool and summarised in the next section

4. GENERAL TUNING METHOD

Although the main goal of this paper is to present the MRPIDLAB, it is necessary to show briefly the general tuning method (GTM) implemented in order to demonstrate the benefits of the tool and the MRPID controller. A more exhaustive description may be found in (Lopez and Cerezo 2007). The most relevant feature of the GTM highlighted in this section is the fact that PID parameters and inner rates are obtained ad hoc using an offline numerical optimization via a heuristic algorithm, simulated annealing (Rutenbar 1989), when classical design methods can not be directly applied or when they lead to a poor performance. As starting point for the optimisation, a solution considering a conventional, single-rate is selected. Thus, the GTM consists of the following steps:

1. Consider a set of specifications with fixed values.
2. Select as period T the maximum of the periods found in the loop.
3. Apply a classical tuning method which allows obtaining control parameters for the single-rate system at period T.
4. Using control parameters obtained in Step 3, minimize a cost function (1) in such a way that specifications can be obtained for the multirate case.

In the last step, from all the obtained parameters, re-tune of K_p, K_i y K_d and/or periods T_p, T_i and T_d is possible through minimizations of a cost function *J*

$$J(K_p,K_i,K_d,T_p,T_i,T_d)=\sum_{i=1}^{m}\beta_i J_i \qquad (1)$$

Since the global control goal may also depend on many different goals, sometimes even contradictory one to another, the relevance of each one can be expressed by weights $\beta_i \in \Re$ with $0 \le \beta_i \le 1$. Besides, any of the goal functions J_i of (1) are:

$$J_i(K_p,K_i,K_d,T_p,T_i,T_d)=\left|\frac{f_s-f_e}{f_e}\right| \qquad (2)$$

being f_e the expected value and f_s the value obtained in the simulation.

As a last remark, this is a local optimization process that generally leads to satisfactory results, but it can not guarantee that the optimal global solution will be achieved.

5. EXAMPLE

This example shows just a bit of the potential that both MRPID controllers and the developed tool have.

In the sequel we will consider a performance requirement is an overshoot M_p=30% at t_p=0.5 seconds and a settling time t_s(2%)<1.7 seconds. The control loop is as follows: the error is measured every T_y=0.2 seconds and the control action is updated every T_u=0.143 seconds, the controller used is a PID with parallel structure and the process model is:

$$G_m(s)=\frac{1}{(0.25s+1)(s+1)} \qquad (3)$$

Four different cases have been tested. The first one considers the problem from the classical discrete single-rate (SR) control theory and imposes a single period T=max(0.2, 0.143)=0.2. With respect to T_h its value is considered in all the experiments equal to 0.001 as imposed by all periods of the global system. Then the real MR problem is dealt with in three different ways: MR-I uses the tuning parameters obtained under the assumption of a single period T, MR-II uses new parameters now considering the multi-rate situation from the beginning, and finally MR-III provides new parameters considering also inner multi-rate. Results have been shown in Tables 1 and 2. In the sequel we give a detailed explanation of the tests.

Table 1: Tuning experiments

Sampling scheme					
Case	T_y	T_u	T_p	T_i	T_d
SR	0.2	0.2	0.2	0.2	0.2
MR-I, MR-II	0.2	0.143	0.2	0.2	0.2
MR-III	0.2	0.143	0.2	0.01	0.143

Table 2: Tuning experiments

Parameters of the controller			
Case	K_c	K_i	K_d
SR, MR-I	4.8459	5.1293	0.8095
MR-II	2.0426	2.9418	1.0559
MR-III	4.4049	5.1806	0.7934

The first one (SR) consists of selecting a sampling period T at which a conventional single-rate tuning is done. Thus considering T=max(T_y,T_u)=0.2, applying pole assignment methods and finally minimising the cost function (4) we obtain the controller parameters K_c, K_i, K_d shown in Table 2.

$$J=\beta_1\left|\frac{M_{ps}-M_{pe}}{M_{pe}}\right|+\beta_2\left|\frac{t_{ps}-t_{pe}}{t_{pe}}\right|+\beta_3\left|\frac{t_{ss}-t_{se}}{t_{se}}\right| \qquad (4)$$

However, if the controller is tuned with such parameters, when considering the real multi-rate situation and T_u drops to 0.143 sec. (MR-I) the performance is degraded. The response of the process and the control action for both SR and MR-I are shown in Fig. 3 and 4 respectively.

Remark 1. SR satisfies the requirements but not the sampling constrains while MR-I satisfies the sampling constrains but, since it uses the same parameters than SR, does not satisfies the requirements. Hence, SR tuning method is not valid when the goal is to tune the controller for MR situations.

The second test (MR-II) minimise (4) again, but now considering the real MR situation from the beginning, with T_r=T_y=T_p=T_i=T_d=0.2 and T_u=0.143. Once the tune is done, Fig. 5 and 6 show the response and the control action with the new results versus the MR-I ones.

Remark 2. The requirements on M_p and t_p are satisfied but not on t_s. Then it would be necessary to have new degrees of freedom in order to make more flexible the controller and thus attain the objective.

In the last test (MR-III) inner multi-rate is added to the controller so that any control action works with different period to another. Thus, having T_p=0.2, T_i=0.01 and T_d=0.143 new parameters are found (see

Table 2). Response and control action are shown in Fig. 7 and 8.

Remark 3. Requirements are now fully satisfied.

Remark 4. It was not possible to do it in any of the former tests. Just when every basic control action took different rates the global control action was able to do it.

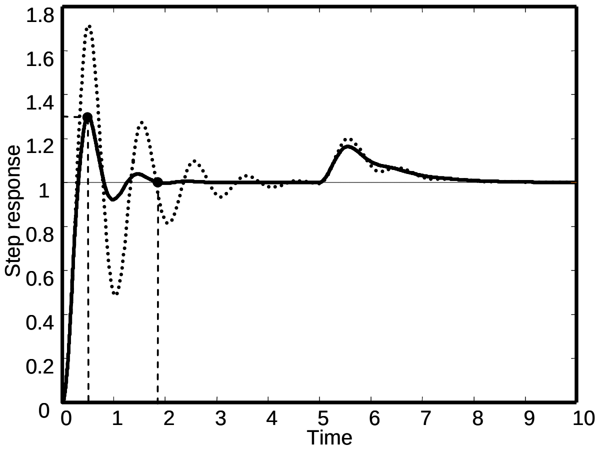


Figure 3: System response for SR (solid) and MR-I (dashed).

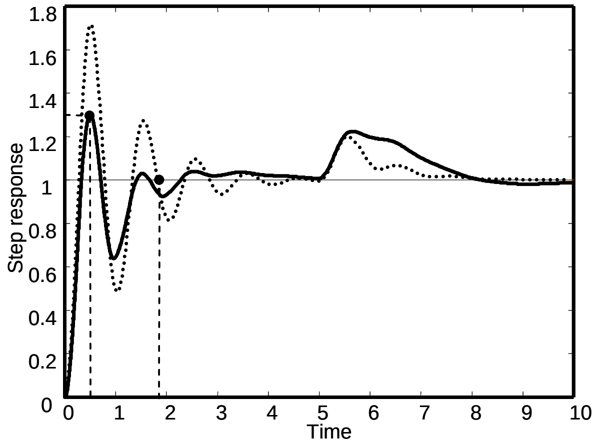


Figure 5: System response with MR-I (dashed) and MR-II (solid).

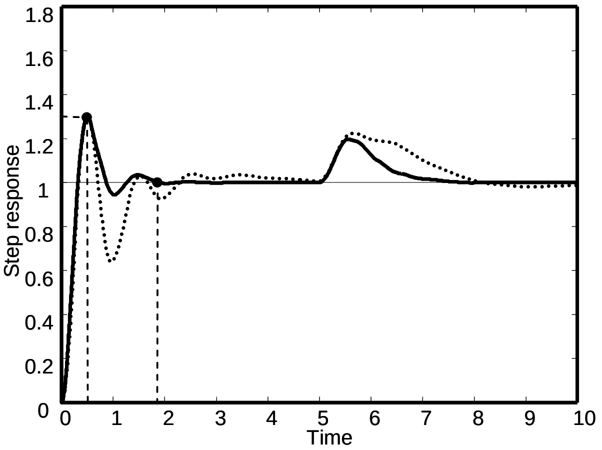


Figure 7: System response with MR-II (dashed) and MR-III (solid).

This is the most relevant result of this batch of tests and justifies the use of the MRPID considering the inner rates as extra degrees of freedom.

Remark 5. Both the tuning method and the simulations are implemented in MRPIDLAB, under a Graphical User Interface that simplify the use.

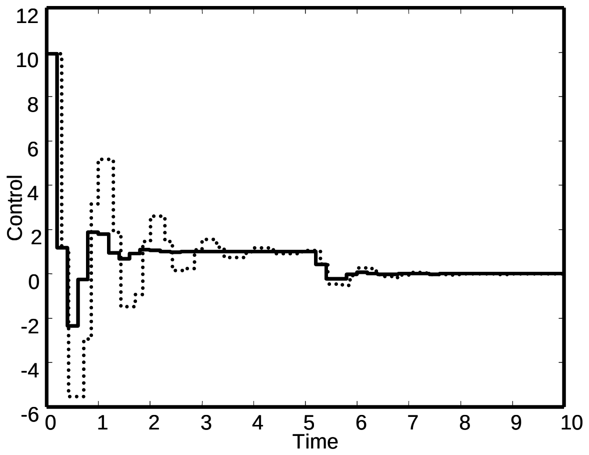


Figure 4: Control action for SR (solid) and MR-I (dashed).

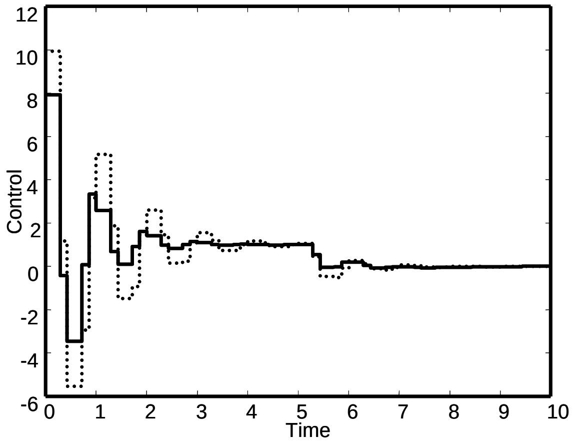


Figure 6: Control action with MR-I (dashed) and MR-II (solid).

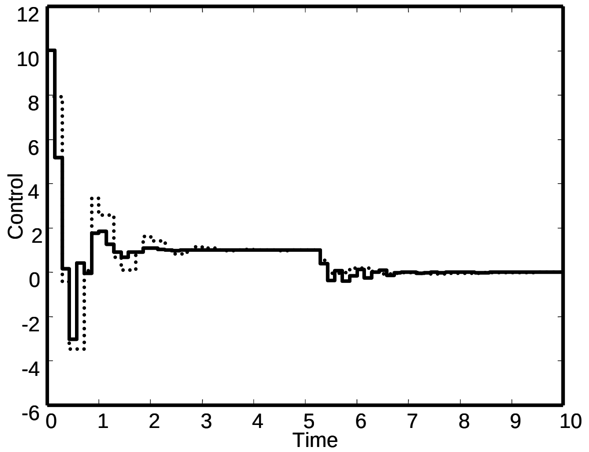


Figure 8: Control action with MR-II (dashed) and MR-III (solid).

6. CONCLUSIONS

The contribution of this paper is to present a new Matlab™, MRPIDLAB. The goal of this tool is twofold: design and simulation. With respect to the former, it takes into account both conventional and non-conventional discrete PID controllers (and continuous when the sampling period is high enough). It is remarkable that non-conventional systems are closer to real problems and that the designing procedure is a multi-objective optimisation. Hence the controller obtained proves a better performance than the single-rate approximation.

On the other hand, the tool serves a simulation environment in which any sampling rate can be set in any variable, different PID configurations can be tested, and saturations, load disturbances, noise measurements and delays can be included.

Finally it is remarkable that the use of MRPID together with the GMT may improve considerably the performance and satisfy the requirements there where conventional PID or tuning methods do not.

ACKNOWLEDGMENTS

This work is carried out in the framework of the research project CICYT. DISCOT. DPI2008-04353 applied by their authors to the Spanish Ministery on Education and Science.

REFERENCES

Albertos, P., Salt, J., Dormido S. and Cuenca, A., 2003. An interactive simulation tool for study of multirate sampled data systems. *IFAC Symposium on Advances Control Education*. Oulu.

Åström, K.J., and Hägglund, T., 2000. The future of PID control. *PID'00, IFAC workshop on digital control,* pp. 19-30. Barcelona.

Åström, K.J., and Hägglund, T., 2005. *Advanced PID control*. Research Triangle Park, NC: Instrument society of America.

Casanova, V., Salt, J., Cuenca, A. and Mascarós, V., 2006. Networked Control Systems over Profibus-DP: Simulation Model. *Proc. of the Int. Conf. on Control Applications*, pp. 1337-1341.

Cuenca, A., 2004. *Modelado, análisis y diseño de sistemas de sistemas de control con muestreo no convencional*. Tesis (PhD). DISA, Universidad Politécnica de Valencia. Spain.

Cuesta, A., Grau, L. and López, I., 2006. CACSD tools for tuning multi-rate PID controllers in time and frequency domains. *IEEE Int. Symposium on Computer Aided Control System Design*. Munich.

Cuesta, A., Grau, L. and López, I., 2007. PID control under sampling period constrains. *Research in Computer Science*. Vol. 31: 63-72.

Franklin, G.F., Powell, J.D. and Workman, M.L., 1994. *Feedback Control of Dynamic Systems*. Addison-Wesley.

González, R., López, I., Morilla, F. and Pastor, R., 2003. Sintolab: the REPSOL-YPF PID tuning tool. *Control Engineering Practice, Pergamon.* pp. 1469-1480.

Gu, Y. and Tomizuka, M., 2000. Digital redesign and Multi-Rate control for motion control – A general approach and application to hard disk drive servo system. *IEEE Advanced Motion Control*, pp. 246-251.

López, I. and Cerezo, Y., 2007. Some practical aspects about performance and tuning of the multirate discrete PID controller. *Proc. of the 15th Mediterranean Conference on Control and Automation*, Atenas (Grecia).

López, I., Dormido, S. and Morilla, F., 1994. The Sampling Period as a Control Parameter. *IFAC Symposium. SICICA, Proceeding of the Congress,* pp. 416-421. Budapest.

Mizumoto, I., Chen, T., Ohdaira, S., Kumon, M. and Iwai, Z., 2007. Adaptive output feedback control of general MIMO systems using multirate sampling and its application to a cart–crane system. *Automatica*, Vol. 43, No. 12: 2077-2085.

Rutenbar, R.A., 1989. Simulated Annealig Algorithm-An overview. *IEEE Circuits and Devices Magazine* 5(1), pp. 19-26.

Sala, A., 2005. Computer control under time-varying sampling period. An LMI gridding-approach. *Automatica*, Vol 41, pp. 2077-2082.

Salt, J. and Albertos, P., 2005. Model-based multirate controllers design. *IEEE Trans. on control systems technology.* Vol. 13 No.6. pp. 988-997.

Salt, J., Cuenca, A., Casanova, V. and Mascarós, V., 2006. A PID dual rate controller implementation over a networked control system. *Proc. of the Int. Conf. on Control Applications,* pp. 1343-1349.

Tomizuka, M., 2004. Multi-rate control for motion control applications. *8th IEEE Intl. Workshop on Advances Motion Control*. pp. 21-29.

Vélez, C. M. and Salt, J., 2000. Simulation of irregular multirate systems. *8th IFAC Symposium on Computer Aided Control System Design.*

Yang, L., and Yang, S., 2007. Multi-Rate control in Internet-Based control systems. *IEEE Trans. on Systems, Man. and Cybernetics-Part C.* Vol. 32. No. 2: 185-192.

EVALUATION AND OPTIMISATION OF MANUFACTURING SYSTEM USING SIMULATION MODELLING AND DESIGN OF EXPERIMENT

Vittorio CESAROTTI$^{(a)}$, **Bruna DI SILVIO**$^{(b)}$, **Vito INTRONA**$^{(c)}$, **Giovanni MORI**

Department of Mechanical Engineering, University of Rome “Tor Vergata”

$^{(a)}$cesarotti@uniroma2.it; $^{(b)}$bruna.di.silvio@uniroma2.it; $^{(c)}$vito.introna@uniroma2.it;

ABSTRACT

The application of computer simulation has been proposed and implemented to optimize an integrated manufacturing system using lean manufacturing principles.

A simulation model only acts as a tool in examining performance. It is essentially a trial and error methodology, and does not directly provide explanations for observed system behaviors. Therefore, in this paper the use of design of experiment in simulation is studied to solve decision-making problems in integrated manufacturing systems. In order to achieve the objectives described above, the authors have developed a simulation model for a manufacturing process in packaging area.

In particular, the authors have been modeling the automatic material handling and storage system served by automatic guided vehicles (AGV) versus packaging lines in a pharmaceutical plant. The lean manufacturing principles have been used to simulate different settings of the process as bottleneck removal, buffer removal and kitting operation introduction. The design of experiment 2^5 factorial design has been used to optimize the scenario.

Keywords: Simulation, Design of experiments, Factorial experiment, AGV, Pharmaceutical Plant

1. INTRODUCTION

The modeling and analysis of integrated manufacturing systems have become more and more important since the wide acceptance of factory automation. However, in many integrated manufacturing systems, production processes are complicated by many interactions between these processes such as deadlock, conflict, as well as uncertainties in the manufacturing environment such as machine failures, tool changes or variability in production requirements. As a result, technologies based on specific management objectives are necessary to model and analyze this class of systems (Tsai 2002).

An approach that can assist engineers and managers is the application of computer simulation (Law 1991). Since the early development of models and languages, simulation has evolved into a technique, which is extremely useful as a facility to test on the model rather than the real-world system, and also to analyze the relationships between the parameters and output behavior. Moreover, there is flexibility in the use of simulation languages, the model can be built as close to reality as we need and taken as a decision-making support tool. Thus, it is very helpful to analyze, schedule or plan manufacturing systems using simulation instead of using complicated mathematical model equations (Galbraith 1994).

Significant work has been performed over the past 25 years in the areas of simulation language development, simulation model design, and model/memory optimization (Tsai 1997). Less attention has been focused, however, on the issues associated with the use of the simulation model as a design and analysis tool. Often, once a simulation model has been verified and validated, the modeler will initiate a series of tests in a random fashion in order to determine the effect of these changes on the model’s output, or response. It is essentially a trial and error methodology, and does not directly provide explanations for observed system behaviors. This approach to scenario creation often results in a “good” solution to the design problem, but does not always result in an optimal solution (Callahan 2006).

Many existing manufacturing system design procedures attempt to minimize a static measure of material handling time or cost, but the performance of a manufacturing system can also depend on other factors such as the batch sizes of parts, scheduling rules, downtimes and setup times on machines, and demand. So, basically there is a need to determine the combination and level of these factors so that a measure of performance is optimized (Ekren and Ornek 2008).

A simulation model only acts as a tool in examining performance. The activities involved in simulation models are to predict results from operational parameters and to select the best solution from a variety of possible options. Combining simulation modeling with design of experiments analysis can be a powerful tool in developing near optimal solutions in a short period of time.

This paper describes a systematic methodology for the use of DOE methods in conjunction with a system simulation study. An application of this methodology is also presented that analyzes a specific manufacturing system of pharmaceutical plant. In this paper, the use of Factorial experiment in the activities of simulation is proposed to achieve the objectives above.

The experimental design is carried out by simulating the system using the ARENA 10.0 (Hammann 1995) simulation software and analyzing outputs using Minitab statistical package

2. USE OF DESIGN OF EXPERIMENT

Experimental design is a strategy to gather empirical knowledge, i.e. knowledge based on the analysis of experimental data and not on theoretical models. It can be applied whenever you intend to investigate a phenomenon in order to gain understanding or improve performance.

Building a design means carefully choosing a small number of experiments that have to be performed under controlled conditions. There are four interrelated steps in building a design (Montgomery 2005):

- Define an objective to the investigation, e.g. better understand or sort out important variables or find optimum.
- Define the variables that will be checked during the experiment (design variables), and their levels or ranges of variation.
- Define the variables that will be measured to describe the outcome of the experimental runs (response variables), and examine their precision.

Among the available standard designs, the modeler choose the more compatible with the aims, number of design variables and precision of measurements, and reasonable cost.

Standard designs are well-known classes of experimental designs. They can be generated automatically as soon as you have decided on the objective, the number and nature of design variables, the nature of the responses and the number of experimental runs you can afford. Generating such a design will provide you with a list of all experiments you must perform, to gather enough information for your purposes.

Design of Experiments (DoE) is widely used in research and development, where a large proportion of the resources go towards solving optimization problems. The key to minimizing optimization costs is to conduct as few experiments as possible. DoE requires only a small set of experiments and thus helps to reduce costs

Design models are very different among authors, particularly in the names of activities and in the level of whom tasks are defined. But the models consistently identify similar types of activities as central to design: problem identification and definition, ideation, evaluation and analysis, and iteration as quintessential examples. Furthermore, most models recognize that design projects transition pass through phases, or alternatively, that designers operate at different cognitive levels of abstraction over the course of a design project. Again, the phases, cognitive levels and labels can differ widely, but most models start with an early conceptual phase, end with a detail design phase, and connect the two with one or more intermediate phases.

Factorial design are widely used in experiments involving several factors whose it is necessary to study the joint effect of the factors in a response.

It is very often required to investigate the effect of several different sets of treatments, or more generally several different explanatory factors, on a response of interest.

Factorial designs allow for the simultaneous study of the effects that several factors may have on a process. Tester make an experiment, varying the levels of the factors simultaneously rather than one at a time is efficient in terms of time and cost, and also allows for the study of interactions between the factors. Interactions are the driving force in many processes. Without the use of factorial experiments, important interactions may remain undetected.

The different aspects defining treatments are conventionally called factors, and there is typically a specified, usually small, number of levels for each factor.

A single treatment is a particular combination of levels of the factors.

A complete factorial experiment consists of an equal number of replicates of all possible combinations of the levels of the factors.

There are several reasons for designing complete factorial experiments, rather than, for example, using a series of experiments investigating one factor at a time. The first is that factorial experiments are much more efficient to estimate main effects, which are the averaged effects of a single factor over all units. The second, and very important, reason is that interaction among factors can be assessed in a factorial experiment but not from series of one-at-a-time experiments.

Interaction effects are important in determining how the conclusions of the experiment might be applied more generally.

Experiments with large numbers of factors are often used as a screening device to assess quickly important effects and interaction. For this reason it is only common to set each factor at two levels, with the aim to keep the size of the experiment manageable. The levels of each factor are conventionally called low and high, or absent and present.

Very often experimenters do not have adequate time, resources and budget to carry out full factorial experiments. If the testers can reasonably assume that some higher-order interactions can be obtained by running only a fraction of the full factorial experiment. A type of orthogonal array design which allows experimenters to study main effects and desired interaction effects in a minimum number of trials is called a fractional factorial design. These fractional factorial designs are generally represented in the form $2^{(k-p)}$ where k is the number of factors and $1/2^p$ represents the fraction of the full factorial 2^k.

This paper takes full advantages of factorial experimental design and simulation to identify and to weight the importance of different factors in the operation of integrated manufacturing system (Antony 2005).

The following methodology is proposed as a technique for quickly and effectively gaining information from a simulation model.

1. Develop a simulation model that addresses the impact of the more important factors in the system on output.
2. Perform verification and validation on the simulation model.
3. Determine the experimental design for a 2k factorial using the inputs and outputs from the simulation model.
4. Adjust inputs to the simulation model in various combinations, as specified by the factorial design, and collect data from the simulation output.
5. Use DOE techniques (and software) to conduct a factorial analysis using the inputs of the simulation as factors and the output of the simulation as the response or responses.
6. Interpret the data and determine the best combination of input factor settings using the ANOVA, effects graphs, and interaction graphs.
7. Repeat the process, if necessary, to further refine significant factor levels.

3. A CASE STUDY OF PHARMACEUTICAL PLANT

This case study is about a pharmaceutical plant and in particular its automated guided vehicles (AGV) system.

The aim was the optimization of AGV's flow between packaging area (called white area) and warehouse

The layout is shown in figure 1. In particular this case study refers to packaging and warehouse areas in order to optimize the AGV flow.

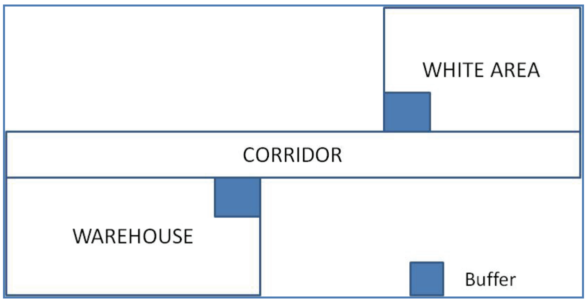


Figure 1: Layout area case of study

The main elements in the system were:

- Warehouse: it has been automated by 5 automated storage and retrieval machines (ASRS) that transport units on 5 roller conveyors. It had a capacity of 15000 cells. Shelving cells are structured for euro-pallets dimension. At the end of the warehouse there were 5 roller conveyors that canalized units versus corridor and packaging area.
- Shape Control: it was an activity to control the shape of the pallets and assigned the cell before storing in the warehouse. There were 3 stations for shape control.
- Buffers: two buffers were present in the layout. The first one was at the end of warehouse in which units are cumulate holding AGV. The second one was before white area, at the end of the corridor, in which units holds to be loading in the packaging lines.
- AGV: two type of AGV were in the considered area.
  The AGV called "White" had the function to transport the pallets through the black corridor. The white AGV were 4 with capacity of 1200 kg each one.
  The AGV called "Blue" had the function to transport as pallets as bins through White Area that was the packaging area of medicinal drugs. The blue AGV were 2 with a capacity of 2200 kg each one.
  Pallets dimensions were 1.160 mm x 1.200 mm x 2.300 mm for both AGV types. Moreover both types was endowed with bumpers and sensors for automatic stopping in case of obstacles presence.

3.1. Modeling Issue

Following the lean manufacturing principles (Melton, 2005), a simulation model was developed to assess:

- removal of the two buffers in order to serve the packaging lines just in time. Resource wastes existed in drop & pick of AGVs in the different buffers. Buffers removal aimed to decrease AGVs cycle time and delay of the packaging lines starting production process due to the absence of processing batch conveyed by AGVs.
- removal of roller conveyors at the end of the warehouse that were a structural bond in the system and bottleneck. The model simulates the absence of roller conveyors and the presence of a pick and drop station at the end of each automated storage and retrieval machines (ASRS) that could be served directly by AGVs.
- optimization tool: this tool was add-on to the model because the AGVs were inclined to come back to the battery charger points after a mission failed according to logic that assigns the missions near the drop stations. The

optimization tool has the task to find the path with the less times for the mission in the whole possible paths. The impact of the time necessary to elaborate the path solution has been assessed with simulation model.
- kitting activity: to avoid the presence of different types of material in a pallet a kitting activity was design at the end of the warehouse. In the pharmaceutical sector laws forbid the presence of incompatible materials in the same pallet. The impact of the presence of kitting activity on the system has been assessed with simulation model.

The simulation model has been validated under different setting conditions. Simulation model design is shown in figure 2.

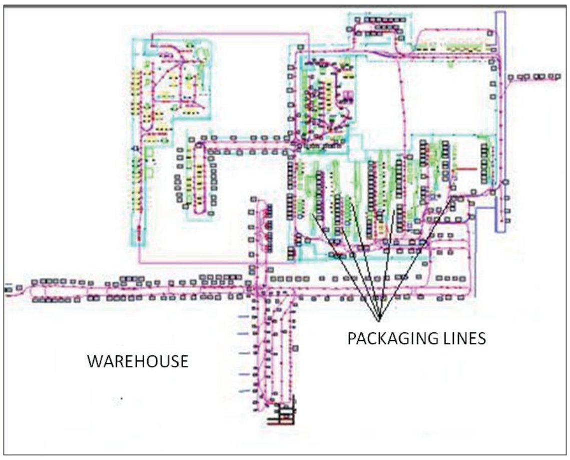


Figure 2: Simulation model design (Arena elaboration)

3.2. Design of Experiment

In the following the steps for the analysis are shown.

3.2.1. Factors and levels

In the design of experiment, process variables include both factors and responses. The selection of these variables is best done as a team effort. The team should:

- include all important factors;
- check the factor settings for impracticable or impossible combinations;
- include all relevant responses;
- avoid using only responses that combine two or more measurements of the process.

From this point of view the project team has identified the variables and their levels for the simulation experiment.

As shown in the previous paragraph, simulation model deals with 5 variables critical for the system:

- contemporaneous lines;
- optimization tool;
- kitting activity;
- number of blu AGV;
- number of white AGV.

The response is the delay of the packaging lines starting production process due to the absence of processing batch conveyed by AGV.

Each variable has been analyzed in order to determine the number of levels in 2^5 DOE design:

- number of blu AGV: this factor, called A, is set in the real system in 2 AGV; in the simulation we are interested to assess the effect of 2 (+) or 1 (-) AGV in action;
- number of white AGV: this factor, called B, is set in the real system in 4 AGV; in the simulation we are interested to asses the effect of 4 (+) or 3 (-) AGV in action;
- kitting activity: this factor, called C, may influence the number of AGV cycles in the system; the levels are relating to the presence (+) or not (-) of this activity;
- optimization tool: this factor, called D, may influence the AGV cycle time in the system; the levels are relating to the presence (+) or not (-) of this tool;
- contemporaneous packaging lines: this factor, called E, influences the cycle time of system; the levels have been fixed to 3 (-) or 4 (+) contemporaneous lines after simulation runs.

3.2.2. DOE: first step analysis

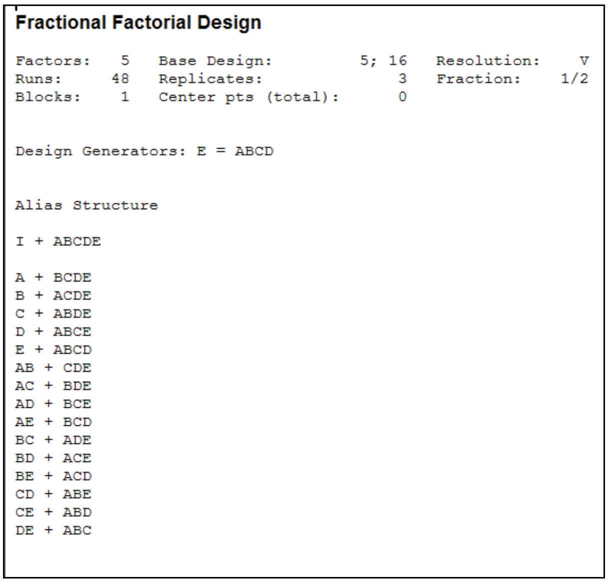
The objectives for the experiments have been determined by a team discussion. First of all it was necessary to identify which factors/effects were important. Factorial design are widely used in experiments involving several factors whose it is necessary to study the joint effect in a response. In this case the number of parameters (5) yields number of experiments equal to 32 (2^5). Referring to the objectives of this experiment is useful to use an half reduction design of 16 (2^{5-1}) experiments.

The experimental design and the alias structure obtained by Minitab software are shown in figure 3.

A confounding design is one where some treatment effects (main or interactions) are estimated by the same linear combination of the experimental observations as some blocking effects. In this case, the treatment effect and the blocking effect are said to be confounded. Confounding is also used as a general term to indicate that the value of a main effect estimate comes from both the main effect itself and also contamination or bias from higher order interactions. They also occur whenever a fractional factorial design is chosen instead of a full factorial design.

In the case it was V resolution design with ABCED = I: no main effect or two-factor interaction is aliased with any other main effect or two-factor interaction, but two-factor interactions are aliased with three-factor interaction.

The higher the resolution, the less restrictive the assumptions that are required regarding which interactions are negligible to obtain a unique interpretation of the data.



```
Fractional Factorial Design

Factors:   5   Base Design:         5; 16   Resolution:    V
Runs:     48   Replicates:              3   Fraction:    1/2
Blocks:    1   Center pts (total):      0


Design Generators: E = ABCD


Alias Structure

I + ABCDE

A + BCDE
B + ACDE
C + ABDE
D + ABCE
E + ABCD
AB + CDE
AC + BDE
AD + BCE
AE + BCD
BC + ADE
BD + ACE
BE + ACD
CD + ABE
CE + ABD
DE + ABC
```

Figure 3: First step DOE (Minitab elaboration)

The result could be discussed after the simulation that takes one hour each one.

Before the conclusions from the first step of DOE, the adequacy of the model should be checked. Factorial design makes assumptions about the errors:

- the errors are normally distributed with mean zero;
- the error variance does not change for different levels of a factor or according to the values of the predicted response;
- each error is independent of all other errors. In a designed experiment, the best way to obtain independent errors is to randomize the run order of the experimental trials.

The diagnostic tool, used in the case study, is residual analysis shown in figure 4.

The probability plot is used for the standardized residuals to check whether:

- the distribution assumption is appropriate;
- the assumption of equal shape (Weibull or exponential) or scale parameter (other distributions) is appropriate.

The probability plot for standardized residuals combines the data to calculate one fitted line, thereby making it easier to determine if the plot points hug the fitted line. If the plot points hug the fitted line then the assumptions are appropriate. The assumptions can be violated when the line does not adequately fit the points. In the case study the plot points hug the fitted line adequately and therefore provide evidence that the assumptions are validated.

A histogram of the residuals shows the distribution of the residuals for all observations. Testers use the histogram as an exploratory tool to learn about the following characteristics of the data:

- typical values, spread or variation, and shape;
- unusual values in the data.

The histogram of the residuals should be bell-shaped.

In the case study the residuals didn’t appear to indicate the presence of an outlier confirming the normal distribution.

Residuals versus fits plots the residuals versus the fitted values. The residuals should be scattered randomly about zero. This plot points out noncostant variance an outlier.

In the case study the residuals appear to be randomly scattered about zero.

Residual versus error graph plots the residuals in the order of the corresponding observations. The plot is useful when the order of the observations may influence the results, which can occur when data are collected in a time sequence or in some other sequence, such as geographic area. This plot can be particularly helpful in a designed experiment in which the runs are not randomized. The residuals in the plot should fluctuate in a random pattern around the center line. Testers examine the plot to see if any correlation exists among error terms that are near each other. Correlation among residuals may be signified by:

- an ascending or descending trend in the residuals
- rapid changes in signs of adjacent residuals

For the case study data, the residuals appeared to be randomly scattered about zero. No evidence seemed to exist that the error terms were correlated each other.

Therefore the validity of these assumptions in analysis was confirmed.

The Pareto Chart, shown in figure 5, pointed out the importance on response (with significant level α=0,01) of all factors except for optimization tool. Moreover the effect of kitting activity, number of blu AGV, number of white AGV was influenced by alias structure. Therefore, the factor optimization tool was set the level of (-), the factor contemporaneous lines was set the level (-) as discussed in the results paragraph.

3.2.1. DOE: second step analysis

A complete factorial design 2^3 has been performed including the remaining variables.

The experimental design obtained by Minitab software is shown in figure 6. All terms are free from aliasing.

The results have confirmed the important effect (with significant level α=0,01) of factors number of AGV blu and white as shown in figure 7. On the other hand, no influence of kitting activity has been foreseen.

The main effects plot, as shown in figure 8, shows the effect of the factors on the response and compare the relative strength of the effects.

The factor number of blu AGV has been set the level of (+) and the factor number of white AGV has been set the level (+) as discussed in the results paragraph.

The factor kitting activities has been set (+1) to evaluating the third combination factors by cube plot as shown in figure 9.

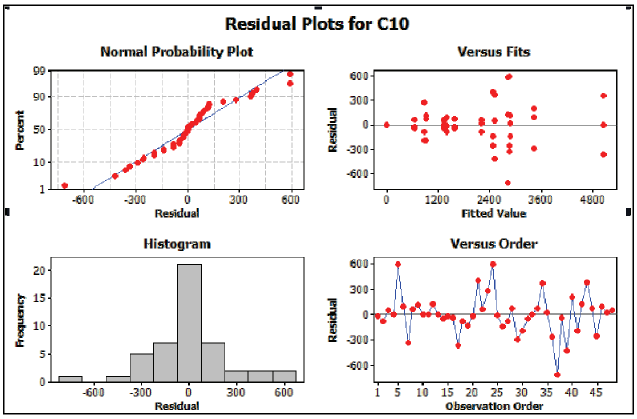


Figure 4: Residual analysis (Minitab elaboration)

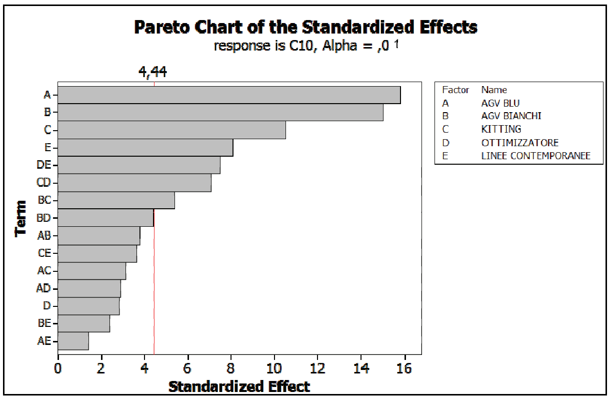
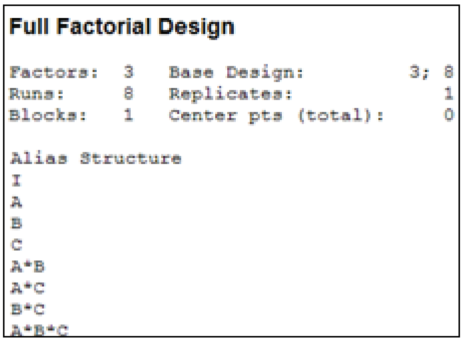


Figure 5: Pareto chart first step DOE (Minitab elaboration)



Full Factorial Design

```
Factors:  3   Base Design:          3; 8
Runs:     8   Replicates:              1
Blocks:   1   Center pts (total):      0

Alias Structure
I
A
B
C
A*B
A*C
B*C
A*B*C
```

Figure 6: Second step DOE (Minitab elaboration)

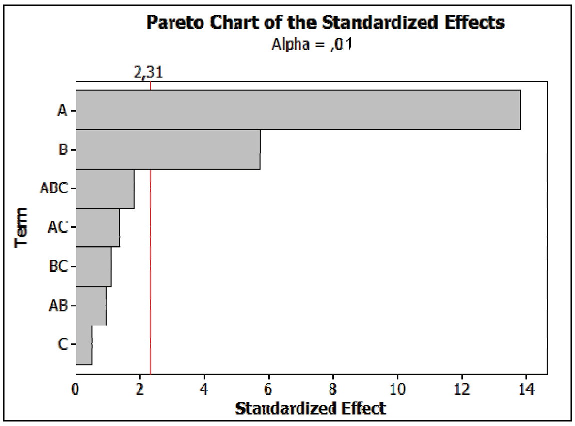


Figure 7: Pareto chart second step DOE (Minitab elaboration)

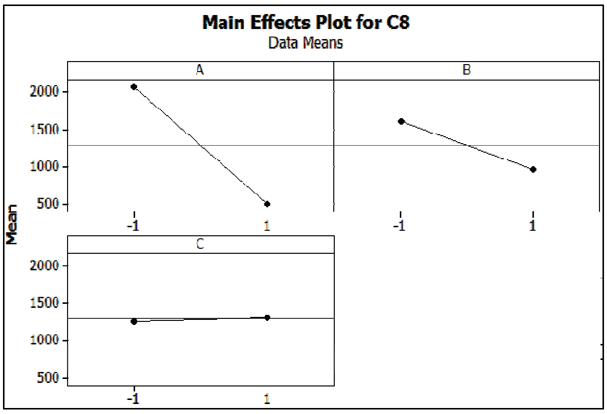


Figure 8:Main effect plot second step DOE (Minitab elaboration)

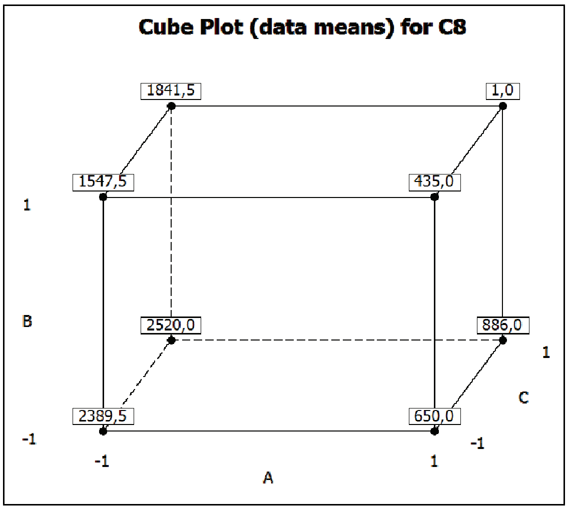


Figure 9: Cube plot second step DOE (Minitab elaboration)

3.2.2. DOE Results

Summarizing, the optimal configuration was:

- Blu AGV set at the number of 2 AGV; it's the most important factor for the response in the system with 1500 s of impact on performance as shown in figure 7; this is due also to the functionality of this type of AGV in the system that may transfer both BIN missions and pallets as shown in the previous paragraph;

- White AGV set at the number of 4 AGV; the running operation of white AGV has an important impact of 500 s on the response;
- Kitting activity set at level (+1) because its presence is requested in the system; but , as shown in figure 7, the absolute value doesn't have an important effect; but considering the combined effect with numbers of AGV, fig.7, factor we could conclude that the presence of kitting activity improve (435 s) the impact on final performance;
- Optimization tool set at level (-) because its presence is not requested in the system; optimization tool improves the AGV flow in the system but slows down the whole performance due to the time for mission allocation;
- Contemporaneous lines set at number of 3. Increasing of contemporaneous lines increases the traffic in the system with an important impact on the final response as shown in figure 9.

The design of experiment has not only allowed to define the optimal configuration with the delay of the packaging lines starting production process equal to 0 s but also to quantify the results of the impact in the system. The optimal configuration allows to reduce the delay of 435 s.

4. CONCLUSIONS

In this paper the application of computer simulation and design of experiment has been proposed and implemented to optimize AGV's flow between packaging area (called white area) and warehouse of a pharmaceutical plant and in particular its automated guided vehicles (AGV) system.

Manufacturing systems can be very complex with numerous paths of material flow and varying capabilities of equipment. Simulation modeling is well suited to represent these systems and can be enhanced by including design of experiments techniques as shown in the case study.

Simulation modeling is well established and can be very useful in analyzing the performance of a manufacturing system. These models can become very complex as the number of factors and system outputs increase. In this article, a study was done to investigate the usefulness of combining DOE techniques with simulation modeling. The DOE techniques were used to determine the number of experiments and factor level combinations necessary to fully and efficiently represent all possible scenarios for the defined model.

REFERENCES

Antony, J., 2005. *Design of experiments for Engineers and Scientists*. Elsevier Butterworth-Heinemann.

Callahan, R. N., Hubbard, K. M.; Bacoski, N. M., 2006. The use of simulation modeling and factorial analysis as a method for process flow improvement. *The International Journal of Advanced Manufacturing Technology*, 29 (1-2), 202 – 208.

Galbraith, L., Standridge, C. R., 1994. Analysis in manufacturing systems simulation: A case study. *Simulation*, 63(6), 368–375.

Hammann, J. E., Nancy, A. M., 1995. *Introduction to ARENA*. *Proceedings of the 1995 IEEE Winter Simulation Conference*, 519–523.

Law, A. M., Kelton, W. D., 1991. *Simulation modeling and analysis*. McGraw-Hill (NY).

Melton T., 2005. The benefits of lean manufacturing: what lean thinking has to offer the process industries. *Chemical Engineering Research and Design*, 83 (A6), 662–673.

Montgomery, D.C., 2005. *Design and Analysis of Experiments*. Wiley &Sons.

Tsai, C., 2002. Evaluation and optimization of integrated manufacturing system operations using Taguchi's experiment design in computer simulation. *Computers & Industrial Engineering*, 43, 591–604.

Tsai, C., 1997. An overview of modeling and methodologies for automated manufacturing system control. *Proceedings of the Tenth Automation Conference*, 240–244.

DATA DRIVEN ADAPTIVE MODEL PREDICTIVE CONTROL WITH CONSTRAINTS

Norhaliza A Wahab(a), **Jonas Balderud**(b) **and Reza Katebi**(c)

Industrial Control Centre, University of Strathclyde, Glasgow, UK

Lisa@eee.strath.ac.uk

ABSTRACT

This paper proposes a Direct Adaptive Model Predictive Controller (DAMPC) with constraints that employs subspace identification techniques to directly identify and implement the controller. The direct identification of controller parameters is desired to reduce the design effort and computational load. The DAMPC method requires a single QR-decomposition for obtaining the controller parameters and uses a receding horizon approach to collect input-output data needed for the controller identification. The paper studies the effect of different horizon schemes, the stability robustness and compares the performance of the proposed control scheme when applied to a nonlinear process with that of a linear model predictive control scheme.

Keywords: Subspace identification, Model predictive control, Adaptive control, Activated sludge process

1. INTRODUCTION

Adaptive controllers are traditionally derived from polynomial transfer function models. This paper demonstrates the use of subspace techniques to provide a state space based adaptive control technique for Model Based Predictive Control (MBPC) design.

Subspace identification techniques have emerged as one of the more popular identification methods for the estimation of state space models from measurement data. Using these techniques, subspace matrices can be constructed and used to obtain prediction of the process outputs. These predictions can subsequently serve as a basis for model predictive controller design. By continuously updating these predictions models an adaptive predictive control method can be obtained.

As an alternative to the two-step adaptive predictive control method that results when a model is explicitly estimated as shown in Fig.1, it is also possible to estimate the control parameters directly from the measurements. This direct adaptive control method was introduced by the adaptive control community in the early 70s (Åström and Wittenmark 1995) and has been widely deployed. Such algorithm combines system identification and control design simultaneously (see Fig.2).

Some previous work has been reported on the design of MPC using subspace matrices such as model-free LQG and subspace predictive controller (Favoreel et al. 1998; Favoreel et al. 1999; Kadali et al. 2003; H.Yang et al. 2005), or using the state space model identified through subspace approach (Ruscio 1997b, c; X.Wang et al. 2007). The main result of (Favoreel et al. 1998, 1999) is that the system identification and the calculation of controller parameters are replaced by a single QR decomposition. Although the idea of combination of subspace methods and MPC has been around for few years, designing an adaptive subspace MPC is still open to discussion. Previous development in subspace based constrained model predictive controllers (Kadali et al. 2002). In H.Yang et al. 2005, for example, considers an adaptive predictive control with sliding window, but does not include the constraints.

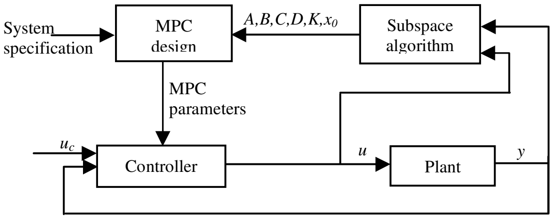


Figure 1: Indirect Adaptive Control

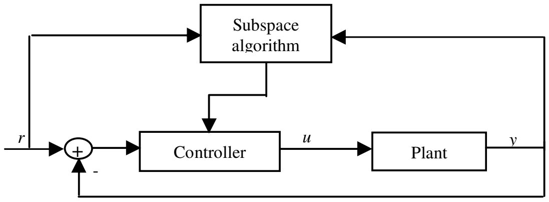


Figure 2: Direct Adaptive Control

Therefore, the objective of this paper is to develop subspace-based adaptive MPC with inclusion of constraint that can cope with mildly nonlinear processes. Other approaches for dealing with these types of processes include linear MPC, nonlinear MPC and neural network based MPC approaches. In practice, however, linear MPC approaches tend to be favoured. Linear MPC approaches include linearization approaches, where a nonlinear model is linearized at each sampling instance (e.g. Krishnan and Kosanovich 1998), and multiple model based approaches (for example see Narenda and Xiang 2000). Previous efforts in the area of adaptive predictive control have also seen the application of neural networks (Wang and J.Huang 2002), however, due to the complexity and computational load typically associated with these methods they have made few inroads in practice.

The proposed adaptive linear MPC method can offer an attractive alternative to existing predictive control methods for mildly nonlinear systems. The proposed method combines the simplicity of linear model predictive control with the power of a self-tuning. The main advantages of the proposed approach are that the usually tedious and time-consuming modelling task can be eliminated and that the controller can adapt to changing process conditions while the physical constraints are satisfied.

The paper is organized as follows: In Section 2 we briefly recapitulate the main concepts of subspace identification and QR-decomposition. The proposed constrained subspace-based MPC approach is developed in Section 3. Section 4 introduces the adaptive MPC. Section 5 describes the application to a wastewater system. The comparison of different control strategies are also presented. The report ends with some conclusion.

2. THE IDENTIFICATION METHOD

A linear discrete time-invariant state space system can be represented as,

$$x(k+1) = Ax(k) + B\Delta u(k) + Ke(k) \tag{1}$$

$$y(k) = Cx(k) + D\Delta u(k) + e(k) \tag{2}$$

where $\Delta u(k)$, $y(k)$ and $x(k)$ are the incremental inputs, outputs and states respectively and where $e(k)$ is a white noise sequence with variance $E[e_p e_q^T] = S\delta_{pq}$. The following matrix input-output equations (De Moor 1988) play an important role in the problem treated in linear subspace identification:

$$Y_f = \Gamma_i X_f + H_i U_f \tag{3}$$

where data block Hankel matrices for $u(k)$ represented as U_p and U_f are defined as:

$$U_p = \begin{pmatrix} u_0 & u_1 & \cdots & u_{j-1} \\ u_1 & u_2 & \cdots & u_j \\ \vdots & \vdots & \ddots & \vdots \\ u_{i-1} & u_i & \cdots & u_{i+j-2} \end{pmatrix} \tag{4}$$

$$U_f = \begin{pmatrix} u_i & u_{i+1} & \cdots & u_{i+j-1} \\ u_{i+1} & u_{i+2} & \cdots & u_{i+j} \\ \vdots & \vdots & \ddots & \vdots \\ u_{2i-1} & u_{2i} & \cdots & u_{2i+j-2} \end{pmatrix} \tag{5}$$

where the subscripts p and f represent ‘past’ and ‘future’ time. The same way, the outputs block Hankel matrices Y_p and Y_f can be defined. i is the prediction (i=H_p) and j is receding window size, n respectively. The extended observability matrix, Γ_i and the lower block triangular Toeplitz matrix, H_i are defined as:

$$\Gamma_i = \begin{pmatrix} C \\ CA \\ \vdots \\ CA^i \end{pmatrix}, H = \begin{pmatrix} CB & 0 & \cdots & 0 \\ CAB & CB & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ CA^{i-1}B & CA^{i-2}B & \cdots & CB \end{pmatrix} \tag{6}$$

The linear predictor equation can now be defined as:

$$\hat{Y}_f = L_w W_p + L_u U_f \tag{7}$$

where, given of past inputs and outputs W_p and future inputs U_f, the problem of subspace identification can be expressed as the solution to the following squares minimisation problem:

$$\min_{Lw,Lu} \left\| Y_f - (L_w, L_u)\begin{pmatrix} W_p \\ U_f \end{pmatrix} \right\|_F^2 \tag{8}$$

The solution to (8) can be found by applying an orthogonal projection of the row space of Y_f into the row space spanned by W_p and U_f as:

$$\hat{Y}_f = Y_f \Big/ \begin{pmatrix} W_p \\ \hline U_f \end{pmatrix} = \underbrace{Y_f /_{U_f} W_p}_{L_w W_p} + \underbrace{Y_f /_{W_p} U_f}_{L_u U_f} \tag{9}$$

The most efficient way to obtain this projection is by applying a QR-decomposition to (9):

$$\begin{pmatrix} W_p \\ U_f \\ Y_f \end{pmatrix} = \begin{pmatrix} R_{11} & 0 & 0 \\ R_{21} & R_{22} & 0 \\ R_{31} & R_{32} & R_{33} \end{pmatrix} \begin{pmatrix} Q_1^T \\ Q_2^T \\ Q_3^T \end{pmatrix} \tag{10}$$

By posing:

$$L = \begin{pmatrix} R_{31} & R_{32} \end{pmatrix} \begin{pmatrix} R_{11} & 0 \\ R_{21} & R_{22} \end{pmatrix}^+ \tag{11}$$

where + denotes the Penrose-Moore pseudo-inverse, equation (9) can be written as:

$$Y_f \Big/ \begin{pmatrix} W_p \\ \hline U_f \end{pmatrix} = L \begin{pmatrix} W_p \\ U_f \end{pmatrix} \tag{12}$$

Then, L_w and L_u can be found from L using (written in Matlab notation):

$$L_w = L(:,1:i(N_u + N_y)) \quad (13)$$

$$L_u = L(:,i(N_u + N_y)+1:i(2*N_u + N_y)) \quad (14)$$

where N_u and N_y denote the number of input and output, respectively.

3. MODEL PREDICTIVE CONTROL METHOD

This section describes the combination of subspace identification and model predictive control. Note that the steps of identification and control design can be carried out simultaneously by applying a single QR-decomposition to the input-output data. This stands in contrast to the design of conventional MPC controllers, where modelling and control design is usually distinctly separated tasks.

The model predictive control problem can in mathematical terms be expressed as the minimization of

$$J = \sum_{i=1}^{H_p} \left(\hat{y}(k+i) - r(k+i)\right)^T Q \left(\hat{y}(k+i) - r(k+i)\right) + \sum_{i=0}^{H_c-1} \Delta u(k+i)^T R \Delta u(k+i) \quad (15)$$

where H_p and H_c denote the prediction and control horizons, respectively. The output and input weighting matrices Q and R are assumed positive definite. By using the linear predictor in equation (7), rewrite the output sequence to include integral action in the predictor:

$$\Delta\hat{y}_f = \tilde{L}_w \Delta w_p + \tilde{L}_u \Delta u_f \quad (16)$$

where $\tilde{L}_w$ and $\tilde{L}_u$ are obtained directly from the previous identification of L_w and L_u while $\Delta w_p = \left[\Delta y_p{}^T \quad \Delta u_p{}^T\right]^T$ and $\Delta\hat{y}_f = \left[\Delta\hat{y}_1, \quad \cdots, \quad \Delta\hat{y}_i\right]^T$. Thus, for a k-step ahead predictor as:

$$\hat{y}_f = y_t + \tilde{L}_w \Delta w_p + \tilde{L}_u \Delta u_f \quad (17)$$

and the current output is:

$$y_t = \left[y_t \quad y_t \quad \cdots \quad y_t\right]^T \quad (18)$$

By substitution of the new integrated linear predictor in equation (17) into the cost function J, differentiate it with respect to Δu_f and equating it to zero gives the control law:

$$\Delta u_f = \left(\tilde{L}_u{}^T Q \tilde{L}_u + R\right)^{-1} \tilde{L}_u{}^T Q \left(r_f - y_t - \tilde{L}_w \Delta w_p\right) \quad (19)$$

Since only the first Δu_f (1) is implemented and the calculation is repeated at each time instant t, therefore given the input $u(t)$ as:

$$u(t) = u(t-1) + \Delta u(t) \quad (20)$$

3.1. Constraints

The following constraints are considered,

$$u_k^{\min} \le u_k \le u_k^{\max}, \; \Delta u_k^{\min} \le \Delta u_k \le \Delta u_k^{\max} \quad (21)$$

Let $u_k = \Delta u_k + u_{k-1}$, then from eq.(21) gives:

$$\begin{aligned} &u_k^{\min} \le \Delta u_k + u_{k-1} \le u_k^{\max} \\ &R\Delta u_k \le u_k^{\max} - u_{k-1} \\ &-R\Delta u_k \le -u_k^{\min} + u_{k-1} \end{aligned} \quad (22)$$

where R is a lower triangular unity matrix. Rewrite the constraint in the incremental inputs as:

$$\Delta u_k \le \Delta u_k^{\max} \qquad -\Delta u_k \le -\Delta u_k^{\min} \quad (23)$$

and then combine those constraints into a single linear inequality:

$$W\Delta u_f \le b \quad (24)$$

where

$$W = \left[-R \quad -I \quad R \quad I\right]^T$$

$$b = \left[-\left(u_k^{\min} - u_{k-1}\right) \quad -\Delta u_k^{\min} \quad u_k^{\max} - u_{k-1} \quad \Delta u_k^{\max}\right]$$

This solution uses a standard QP optimization problem. QP is applied at every instant such that:

$$\begin{aligned} \min_{\Delta u} J &= \left(r_f - \hat{y}_f\right)^T Q \left(r_f - \hat{y}_f\right) + \Delta u_f{}^T R \Delta u_f \\ &= \left(r_f - F - \tilde{L}_u \Delta u_f\right)^T Q \left(r_f - F - \tilde{L}_u \Delta u_f\right) + \Delta u_f{}^T R \Delta u_f \\ &= \Delta u_f{}^T \left(\tilde{L}_u{}^T Q \tilde{L}_u + R\right) \Delta u_f - 2\tilde{L}_u{}^T Q \left(r_f - F\right) \Delta u_f \\ &= \frac{1}{2} \Delta u_f{}^T \Phi \Delta u_f - \phi \Delta u_f \end{aligned} \quad (25)$$

where $\Phi = \left(\tilde{L}_u{}^T Q \tilde{L}_u + R\right)$ and $\phi = -2\tilde{L}_u{}^T Q \left(r_f - F\right)$

4. DIRECT ADAPTIVE MODEL PREDICTIVE CONTROL

This section considers the online implementation of the subspace based model predictive controller. Two types of controllers are considered; one in which the subspace identification data is collected over a sliding (receding) window and one where the identification data is collected in batches.

4.1. Sliding (receding) window

The procedure of using a sliding window for identification is illustrated in Fig.3. The main advantage of this approach is that the controller parameters are updated each sample, which usually means a quicker response to process changes. The main drawback of this method is that a QR-decomposition needs to be computed each sample instance which increases the computation load.

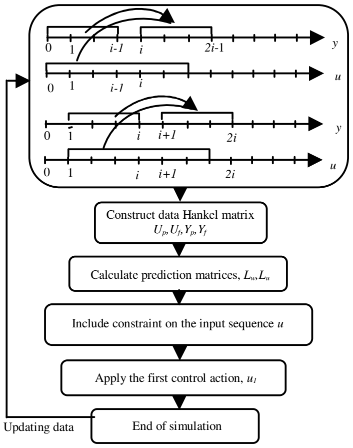

0
1
i-1
i
2i-1
y
0
1
i-1
i
u
0
1
i
i+1
2i
y
0
1
i
i+1
2i
u
Construct data Hankel matrix
U_p, U_f, Y_p, Y_f
Calculate prediction matrices, L_w, L_u
Include constraint on the input sequence u
Apply the first control action, u_I
Updating data
End of simulation

Figure 3: Sliding window for $i=4$

4.2. Batch window

In this windowing approach the control parameters are updated every n^{th} sample, where n is size of the identification window. Since a QR-decomposition is only needed every n^{th} sample the computational load requirements is lower for this method. The main drawback is that the method generally will respond slower to changing process conditions.

5. SIMULATION STUDY

To benchmark the proposed direct adaptive MPC control technique, it has been applied to an activated sludge processes. This process is comprised of an aerator and a settler as shown in Fig.4. The bioreactor includes a secondary clarifier that serves to retain the biomass in the system while producing a high quality effluent. Part of the settled biomass is recycled to allow the right concentration of micro-organisms in the aerated tank. A component mass balance that yields the following set of nonlinear differential equations was previously derived (Takács, I. et al. 1999)

$$\dot{X}(t)=\mu(t)X(t)-D(t)(1+r)X(t)+rD(t)X_r(t) \tag{26}$$

$$\dot{S}(t)=-\frac{\mu(t)}{Y}X(t)-D(t)(1+r)S(t)+D(t)S_{in} \tag{27}$$

$$\dot{C}(t)=-\frac{K_O\mu(t)}{Y}X(t)-D(t)(1+r)C(t) +K_{La}(C_S-C(t))+D(t)C_{in} \tag{28}$$

$$\dot{X}_r(t)=D(t)(1+r)X(t)-D(t)(\beta+r)X_r(t) \tag{29}$$

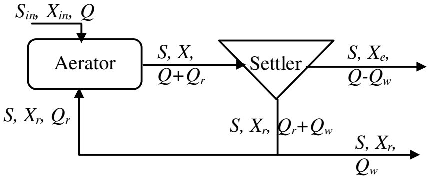


Figure 4: Activated Sludge Reactor

where the state variables, $X(t)$, $S(t)$, $C(t)$ and $X_r(t)$ represent the concentrations of biomass, substrate, dissolved oxygen (DO) and recycled biomass respectively. $D(t)$ is the dilution rate, while S_{in} and C_{in} correspond to the substrate and DO concentrations of influent stream. The parameters r and β represents the ratio of recycled and waste flow to the influent flow rate, respectively. The kinetics of the cell mass production is defined in terms of the specific growth rate μ and the yield of cell mass Y. The term K_o is a constant. C_s and K_{La} denote the maximum dissolved oxygen concentration and the oxygen mass transfer coefficient, respectively. The Monod equation gives the growth rate related to the maximum growth rate, to the substrate concentration, and to DO concentration:

$$\mu(t)=\mu_{\max}\frac{S(t)}{K_s+S(t)}\frac{C(t)}{K_c+C(t)} \tag{30}$$

where μ_{max} is the maximum specific growth rate, K_s is the affinity constant and K_c is the saturation constant. In this simulation, two controlled outputs substrate (S) and DO and two manipulated inputs dilution rate (D) and airflow rate (W) are considered.

5.1. The prediction horizons

The prediction horizon and control horizons that have been employed in the simulations are H_p=35 and H_c=5. The prediction horizon weakly related to the length of the identification window, which in turn is directly related to the computational load of the controller (size of the QR decomposition). The computation time for batch window is 15 minutes, which is 5 minutes faster than sliding window. This is reasonable since that of sliding window updates the controller at each sample time, whilst batch updates at every n^{th} sample. A trade-off between the length of the prediction horizon and the computational load of the controller must therefore be employed. Moreover, the accuracy of predictor depends on the choice of prediction horizon length. In this instance the 'best' prediction horizon length was found by fixing the length of the identification window to n=400. Then several different prediction horizons where benchmarked and it was eventually found that H_p= 35 provided for the best performance. The weighting matrices were tuned using a trial and error approach, and eventually chosen as $Q=diag\{10,1000\}$ and $R=diag\{1,1\}$.

Simulations were carried out for three different control strategies. The first two methods are those proposed in this paper, using a sliding identification window and batch identification window. The third strategy employs a non-adaptive linear MPC controller.

The simulation ran from a steady-state operating point at outputs S=41.23mg/l, D=6.11mg/l and inputs D=0.08 1/h and W=90m^3/h. The set point given for the outputs were allowed to vary approximately 10% around the system's steady state condition. The constraint on the input was given as $0.02\le u_1\le 0.15$ and $0\le u_2\le 300$ whilst constraint on the input changes were allowed to $-0.01 \leq \Delta u_1 \leq 0.01$ and $-5 \leq \Delta u_2 \leq 5$. Fig.5 shows the setpoint tracking for sliding window approach under different constraints on the input change, Δu. For the sliding window₁ is $|\Delta u_1| \leq 0.01; |\Delta u_2| \leq 5$, whilst sliding window$_2$ is $|\Delta u_1| \leq 0.001; |\Delta u_2| \leq 3$. It can be seen that smaller the magnitude of the maximum allowed input changes, more sluggish is the controller response to setpoint changes. Fig.6 compares the setpoint tracking performance of the three control strategies with the input constraints denoted above and constraints on the input changes, ($|\Delta u_1| \leq 0.01; |\Delta u_2| \leq 5$). The sliding window approach converges quickly whilst the batch approach takes somewhat longer to converge. The proposed sliding window algorithm also demonstrates less interaction than the other two and good tracking properties. The linear MPC shows an overshoot to setpoint change.

Fig.7 shows the performance of control when the measurements of substrate and DO are corrupted with step input (Amp=0.001) disturbance at t=1750. It can be seen that the performances deteriorate then able to track back to the setpoint quickly. The linear MPC shows large peak of disturbance on substrate measurement compare to the subspace MPC. Though batch window present the lowest interaction on substrate output due to disturbance, it converge slowly compare to sliding one. The measurement of DO given by sliding window and linear MPC was slightly same compared to batch window which is much slower. The stability test has been performed by simulation. The open loop poles are 0.1360;0.9924;0.8180;0.7727. Table 1 evaluated the closed loop poles for different levels of setpoint in substrate measurement. (see Fig.7)

Table 1: Closed loop poles for a different setpoint

SP A	0.1769	0.4898	0.9885	0.8297
SP B	0.2217	0.3087	0.985	0.8297
SP C	0.1552	0.5273	0.9884	0.8290

It can be seen that the closed loop system remains stable for the setpoint changes. Similar conclusions can be obtained as in the second output DO.

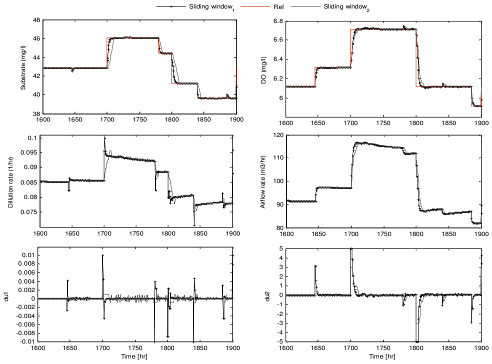


Figure 5: Different constraint case on input changes

6. CONCLUSION

In this paper, the design of model predictive controller from subspace matrices in a framework of adaptive controller is addressed and successfully applied to an activated sludge process. The subspace based model predictive controller using sliding window approach is shown to be more efficient and robust for both outputs than that of batch approach and a linear MPC controller when applied to a mildly nonlinear process.

ACKNOWLEDGMENT

The work has been supported financially by Malaysian Government and Malaysia, University of Technology. This support is gratefully acknowledged.

REFERENCES

D.D. Ruscio., 1997a. Model based predictive control: An extended state space approach. *Proceedings of the 36th Conference on Decision and Control*, San Diego, CA.

D.D. Ruscio., 1997b. Model predictive control and identification: A linear state space model approach. *Proceedings of the 36th Conference on Decision and Control*, San Diego, CA.

De Moor B., 1988. *Mathematical concepts and techniques for modelling of static and dynamic systems.* PhD thesis, Department of Electrical Engineering, Katholieke Univeriteit Leuven, Belgium.

H. Yang, and Li. S, 2005. Subspace-based adaptive predictive control of nonlinear systems. *International Journal of Innovative Computing, Information and Control,1* (4), 743-753.

Kadali, R., B. Huang and A. Rossiter., 2003. A data driven subspace approach to predictive controller design, *Control Engineering Practice*, 7 (3), 261-278.

Krishnan, K., and K.A.Kosanovich, 1998. Batch reactor control using a multiple model-based controller design, *Canadian Journal of Chemical Engineering*, 76, 806-815.

Karl J. °Astr¨om and Bj¨orn Wittenmark, 1995. *Adaptive control*, second ed., Addison Wesley.

Maciejowski J.M., 2002. *Predictive Control with Constraints*. Prentice Hall, England.

M. Verhaegen, 1994. Identification of the deterministic part of mimo state space models given in innovation form from input-output data, *Automatica* 30 (1), 61-74.

Narenda, K.S. and C.Xiang., 2000. Adaptive control of discrete-time system using multiple models, *IEEE Transactions on Automatic Control*, 45 (9), 1669-1686.

P.V. Overschee and De Moor B. 1996. S*ubspace identification for linear systems*, Kluwer Academic Publishers.

P.V. Overschee and De Moor B, 1994. N4SID: Subspace algorithms for the identification of combined deterministic-stochastic systems, *Automatica* 30 (1), 75-93.

Takács, I.,G.G.Patry and D.Nolasco., 1991. A dynamic model of the clarification-thickening process. *Water Research* 25(10), 1263-1271.

W.Favoreel, B. De Moor and M.Gevers., *1999.* Subspace Predictive Control. *Proceedings of the 14th IFAC*, Beijing , China.

W.Favoreel, B. De Moor, P. Van Overschee, and M.Gevers., 1998. Model-free subspace based LQG design. *Proceedings of the American Control Conference*, San Diego, CA.

Wang, D. and J. Huang., 2002. Adaptive neural network control for a class of uncertain nonlinear systems in pure-feedback form, *Automatica*, 38 (8), 1365-1372.

X. Wang, B.Huang, and T. Chen, 2007. Data driven predictive control for solid oxide fuel cells, *Journal of process control* (17), 103-114.

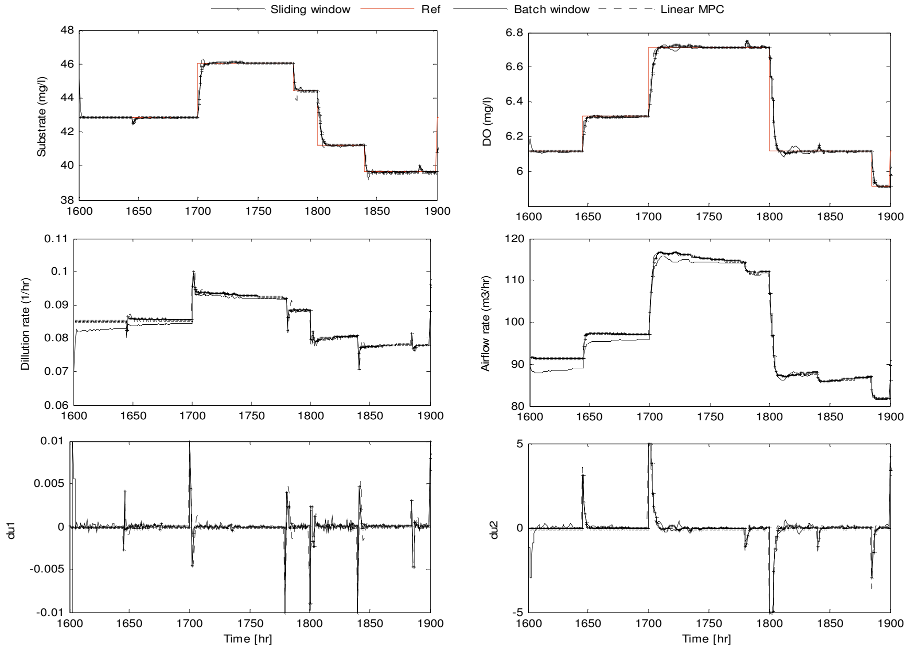


Figure 6: The comparison of control performance (set point tracking)

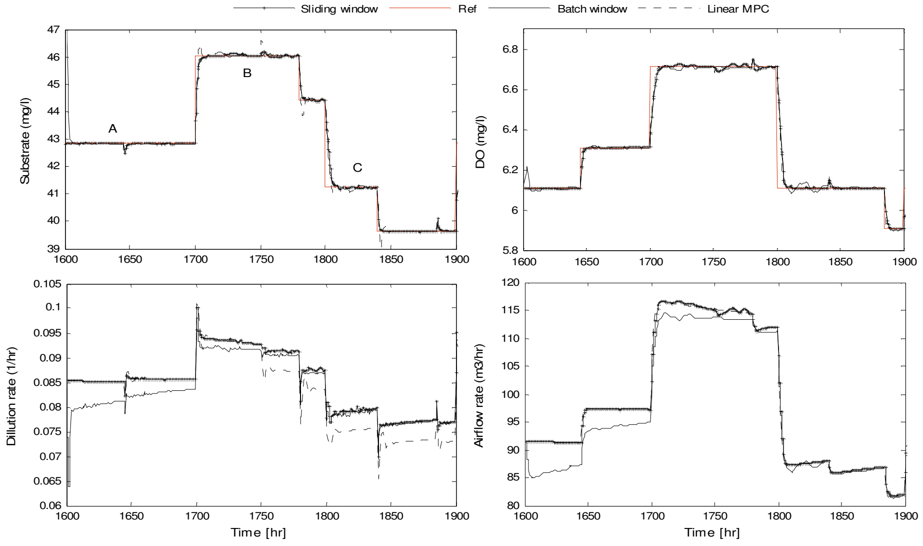


Figure 7: The comparison of control performance with input disturbance at t=1750

PEDESTRAIN CELLULAR AUTOMATA AND INDUSTRIAL PROCESS SIMULATION

Alan Jolly[(a)], Rex Oleson II[(b)], D.J. Kaup[(c)]

[(a,b,c)] Institute for Simulation and Training, 3100 Technology Parkway, Orlando, FL 32826
[(c)] Mathematics Department, University of Central Florida, Orlando, FL 32816-1364

[(a)]brooksjolly@gmail.com, [(b)]quazerin@juno.com, [(c)]kaup@mail.ucf.edu

ABSTRACT

Individuals base decisions on their surroundings and change their minds based on the action of others and interactions with the environment. Current Cellular Automata techniques for modeling pedestrian movement predetermine the individual's goals and do not change them throughout the execution of the simulation. To allow the individuals to make decisions as the simulation progresses a technique has been developed to separate the decision making process of environmental factors and the static environmental effects. Any individual can modify where they are going, based on the locations of other individuals, their capabilities of movement, current velocity and relationships to the environment. This technique will allow individuals to pick the best paths based on what is actually happening during the simulation. This technique can be used to optimize work flow strategies and seek the best way to deal with work stoppages and other problems which may arise.

Keywords: crowd simulation, cellular automata,
walking worker production design

1. INTRODUCTION

The purpose of this paper is to provide a 'proof-of-concept' that a cellular automata (CA) model of pedestrian motion can be integrated into manufacturing job shop production simulations. The outline of this paper is first an introduction to models of pedestrian motion, second a description of the CA model that is used during this research, and third an argument for models of pedestrian motion in job shop production models. Next a brief overview of the modifications we have made to the original CA model which allows the model to be applied to a broad range of pedestrian modeling problems. Finally, we present a simple example of a combined pedestrian-job shop simulation.

2. PEDESTRIAN MODELS

Recently, a considerable amount of research has been done on simulating collective behavior of pedestrians in the street or people finding their way inside a building or a room. Models of crowd behavior attempt to describe collective pedestrian behaviors that result from complex interactions among the individuals composing a crowd and between these individuals and their physical environment. Reviews of the state of the art can be found in a volume containing Schadschneider (2002) and Kessel, Klüpfel, Wahle, and Schreckenberg (2002).

Existing models can be broadly separated into the following two categories: (1) discrete-space models and (2) continuous-space ones. Discrete-space, or cellular automata-based models allow pedestrians to be located at nodes of a fixed or adaptive grid, and pedestrian coordinates are updated at discrete time intervals. Particular models of this category are described in Schadschneider (2002); Blue and Adler (2002); Dijkstra, Jesurun, and Timmermans (2002); Kessel, Klüpfel, Wahle, and Schreckenberg (2002); and Batty, DeSyllas, and Duxbury (2002). The models of the second category allow pedestrians to move continuously in a part of the 2-D surface representing a street, a room, and so forth.

The continuous space models can further be subdivided. Some models, such as the ones considered in Helbing, Farkas, and Vicsek, (2000) and AlGadhi, Mahmassani, and Herman (2002), are based on a similarity between the dynamics of a crowd and that of a fluid or gas. Other models of the second category allow pedestrians to choose their paths by optimizing a certain cost function (Hoogendoorn, Bovy, and Daamen 2002). An interesting model combining the fluid dynamics approach with that of a cost function is considered in Hughes (2002); there the role of the cost function is played by the pedestrian's estimated travel time. Finally, the model considered in other sources (Helbing and Molar 1995, Helbing, Farkas, and Vicsek 2000) introduces social and physical forces among pedestrians and then treats each pedestrian as a particle abiding the laws of Newtonian mechanics.

Generally all the models mentioned above are microscopic and rooted in the 'generalized behavior concept' that is, for a model of crowd movement to be reliable it need not model any certain individual correctly - just the average behavior of a group of individuals responding to certain situations. It is assumed that since pedestrians face similar movement and route choices everyday their response become in effect automatic and can therefore be predicted. In order to mathematically model pedestrian movement one must suppose that an individual's behavior will exhibit certain regularities. With the acceptance of the premise that pedestrians exhibit regular behaviors general mathematical models of crowd dynamics can be formed.

2.1. Cellular Automata and Pedestrian Simulation

This paper focuses on modification to the implementation of Schadschneider's (Schadschneider 2002) cellular automata model of pedestrian dynamics. Cellular Automata (CA) models use logical rules and stochastic processes to define pedestrian motion. CA models divide a pedestrian's movement space into a defined number of identically sized cells, usually in a rectangular grid. Each grid cell has a finite number of states; with pedestrian models this is generally occupied or unoccupied. At a specified time step (generation) the state of the each cell is determined by some function using inputs from a neighborhood of surrounding cells.

Pedestrian movement over the grid is represented by either by sizing the lattice to allow for only one individual per cell or by tracking cell density. Movement between cells is based on a matrix of preferences (transition rules) that determines the state of individual grid cells at each generation. The matrix of preferences and neighborhood of cells are processed by a global transition function that defines cell state, the same function applies to all cells in the lattice. CA models have the advantage of being able to simulate the dynamics of large crowds in less than real time due to their discrete nature.

2.2. Floor Field Approach to CA

Schadschneider (2002) proposed a CA floor field approach that models pedestrians as elementary particles, fermions. They react to their immediate neighborhood with long-range interactions modeled through the use of mediating particles called bosons. In particle physics all elementary particles are composed of either bosons or fermions - fermions (pedestrians) resist being placed near each other and bosons (virtual traces) do not. Pedestrians are fermions thus unable to occupy the same cell (hard-core exclusion principle) and bosons model the virtual traces left behind by the pedestrians as they move over the grid. More than one boson is able to occupy the same cell, so the virtual trace and pedestrians are handled separately.

Pedestrian 'intelligence', by which we mean their choice of movement direction, is modeled through the use of floor fields. Each grid cell has attributes associated with either a dynamic or static floor field. The dynamic floor field changes with each time step as a function of the density and diffusion of bosons. The static floor field remains constant and represents and defines the attraction to exits and the location of obstacles.

The general formula for this CA model is

$$p_{ij} = NM_{ij} \exp(k_d D_{ij}) \exp(k_s S_{ij})(1 - n_{ij}) \qquad (1)$$

In this formula

1. p_{ij} is the probability the pedestrian will move to a neighboring cell.
2. N is a normalization factor ensuring that $\sum p_{ij} = 1$.
3. M_{ij} is the pedestrian's matrix of preferences.
4. D_{ij} is the dynamic floor field value.
5. S_{ij} is the static floor field value.
6. $n_{ij} = 1$ if the cell is occupied.
7. k_s and k_d are coupling factors for the floor field.

The sequence of updates for this model is as follows:

1. Update the dynamic floor field based on diffusion and decay rules.
2. Calculate transitional probabilities (p_{ij}) for each pedestrian.
3. Choose pedestrian's target cell.
4. Resolve conflicts if two pedestrians target the same cell.
5. Execute pedestrian movement.
6. Alter dynamic floor field based on rules (i.e. dropping bosons).

3. CURRENT RESEARCH

Though designed to simulate crowd dynamics we believe models of pedestrian motion can be applied as valuable additions to simulations of other processes. In particular we feel that inclusion of explicit models of pedestrian motion may be beneficial in simulations of manufacturing processes such as walking worker production lines. Simulating worker movement may result in more realistic production output estimates and provide greater insights into human factors that affect the production process.

3.1. Need for Pedestrian Modeling in Industrial Manufacturing

Simulations for job shop performance and layout have traditionally been solved mathematically as 'static' problems. This allows for optimization techniques to be applied to production scheduling and job shop layout problems. In reality job shops operate as dynamic systems with complex interactions between workers and machines (MacCarthy 2001). Patterns of worker movement, the impact of shop-floor layout (local and global configurations) and presence of other workers in the manufacturing process have rarely been explored in job shop simulations.

Walking worker production designs may potentially benefit from explicit simulation of worker (pedestrian) movement. Under this method workers build a product completely from start to end. Walking workers production designs provide flexibility in capacity as workers may be added or removed from the production line in response to the output demand. Two common types of walking worker production lines are: 1) liner production lines where assembly operators travel along the line carrying out each assembly task at each workstation, and 2) fixed-position assembly lines where products are placed at fixed position workstations and assembly operators move between workstations.

Past simulations of walking worker production lines have assumed equal worker efficiency and movement times (Wang 2005). Celano, Costa, Fichera, and Perrone (2004) used the critical worker concept and parameterized walking speeds to obtain more realistic simulation results for linear walking worker production line scenarios. A critical worker is one who does not have enough space or time to complete their task within a workstation thus bringing production to a halt (Celano, Costa, Fichera, and Perrone, 2004). Workers nearby who have free time and the correct skills are able to help critical workers and reduce stoppage times; however, simulation of this requires knowledge of worker location and thus the explicit modeling of worker positions. The worker's position and walking speed within their workstation influences the possibility of their intervention to help a critical operator during a stoppage.

Position, movement, and route choices of the assembly operators also play a major role in the simulation of fixed workstation production lines. Figure 1 shows a process diagram for fixed workstation walking worker models.

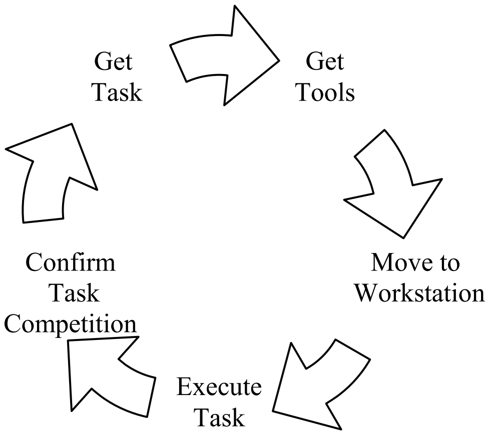


Figure 1 – Process Diagram for Assembly Operators

At a minimum two of the process tasks (get tools, move to workstation) are significantly impacted by human factors of movement and are highly dynamic. Inclusion of pedestrian movement models may also help identify bottlenecks caused by human factors association with position and movement, test the robustness of the production line design to handle the addition and subtraction of workers, and assist in the design of the production line, for example moving workstations that are likely to have critical worker stoppages to highly trafficked areas.

3.2. Modifications to Pedestrian Model

Our implementation of Schadschneider's (2002) floor field CA model modifies the static floor field by splitting it into two fields: 1) one static field for obstacles and static environmental attraction and repulsion forces and 2) a second dynamic field which determines individual's attraction towards a goal or point of interest for the individual. Additionally we deviate from their homogeneous approach by allowing each agent to store its own representation of the dynamic field. This allows each individual pedestrian to change desired destinations based on the evolving environmental conditions.

Through this implementation individuals can be seen to adjust their movement and choose a new exit when conditions such as their inability to move forward, or the density at an exit becoming too large. Modifications of our technique to allow individuals to dynamically select exits based on environmental conditions allows for production line workers to shift between tasks and workstations as well as to go to workstations which need the most assistance.

Cellular automata models as discrete models are well suited for the inclusion of hierarchal rules sets that provide 'intelligence' to the simulated agents. Using multiple static floor fields to describe the job-shop environment and use of hierarchal rule sets to determine agent (worker) objectives, we demonstrate a 'proof-of-concept' that models of pedestrian motion can be integrated into production process simulations. We do this by example using standard linear and fixed position job shop layouts for walking worker production lines. Our production line simulations are implemented in the UCF Crowd Simulation framework (available at http://www.simmbios.ist.ucf.edu/Research/DynamicHumanBehaviors/Repository.aspx) built using the MASON library (Luke, Cioffi-Revilla, Panait, Sullivan and Balan, 2005).

4. PRELIMANRY RESEARCH

4.1. Comparison Job Shop Model

The manufacturing system design used in this research was presented as an example of a Job Shop model in Law and Kelton (2000). The job shop is modeled as a network of five-multiserver queues. The system has five workstations with 3,2,4,3 and 1 identical machine(s) respectively. Job inter-arrival times are identically individually distributed (IID) exponential random variables with a mean of 0.25 hour. There are three job types that arrive with probability 0.3, 0.5, and 0.2. The jobs must be completed in a certain route order. Table 1 shows the number of tasks and routing for each job type.

Table 1: Job tasks

Job Type	Workstation routing
1	3,1,2,5
2	4,1,3
3	2,5,1,4,3

If a job arrives at a workstation and all machines are in use the job joins the first-in-first-out queue for that workstation. The time to perform the individual tasks that make up the overall job is an independent 2-erlang random variable with a mean based on job type and station. Table 2 two defines the mean service times for tasks by workstation.

Table 2: Task Service Times (Hours)

Job Type	Mean service times for tasks
1	0.50, 0.60, 0.85, 0.50
2	1.10, 0.80, 0.75
3	1.20, 0.25, 0.70, 0.90, 1.00

Law and Kelton (2000) used discrete-event simulation to model the system with arrival, departure, and end of simulation comprising the three possible event types. The code to run this model was copied from Law and Kelton (2000) and is written in C. A significant feature of their implementation is that once a workstation task is completed the job is instantaneously transmitted to the next workstation to begin work or be added to that workstations queue.

4.2. Pedestrian inclusion in Job Shop Model

Law and Kelton's (2000) job shop model was implemented in our crowd simulation framework so that we may explicitly model a walking workers movement as part of the job shops manufacturing process. A simplified version of Schadschneider's (Schadschneider 2002) floor field model for pedestrian motion (described in section 2.2) was used to control worker motion. The matrix of preference and dynamic floor field were excluded.

The matrix of preference is unnecessary as our crowd simulation framework recalculates the static floor field as waypoints change and the worker's preference is always to go towards the waypoint. The matrix of preference is generally used to override the static and dynamic floor fields when, for example, individuals 'get-stuck' during evacuation simulations. The dynamic floor field is used to model the tendency of individuals to follow one another and allows for the characteristic lane formation seen in crowd dynamics. For this application we felt this tendency to follow your predecessor's footsteps was not needed. The individuals are primarily driven by the location of the next workstation (waypoint). Equation 2 shows the general formula for the equation of motion used in the pedestrian job shop model.

$$p_{ij} = N \exp(k_s S_{ij})(1 - n_{ij}) \qquad (2)$$

We assume that there are always enough workers to associate with arriving jobs and the distributions and means for job inter-arrival time and task service time are the same as those given in the Law and Kelton model (2000).

The workflow for our JAVA implementation of the model for individuals is shown in Figure 2.

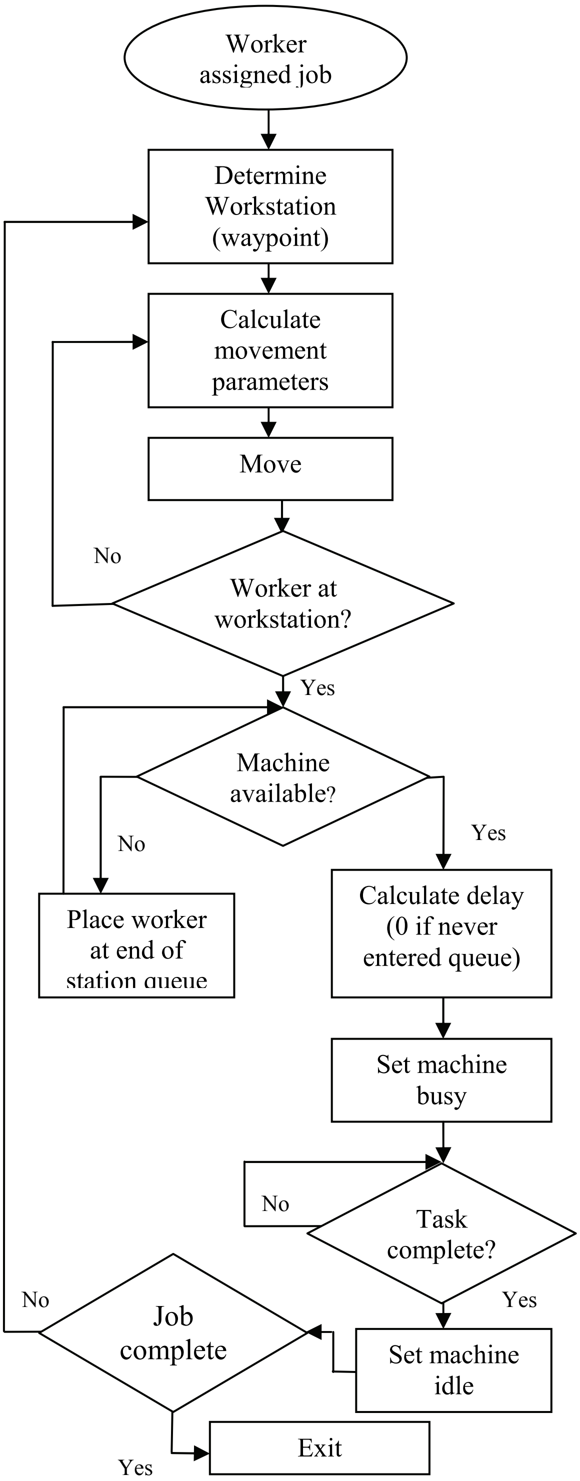


Figure 2: Flowchart for Worker Process

We do not queue workers by having them stand in line. Getting agents to self organize and maintain a formation is a problem not currently addressed in our crowd simulation framework. Workers entering the queue are relocated to a cell just below the workstation (the queues are labeled in Fig. 3). This is the only time that more than one individual is able to share the same cell. When a worker is removed from the queue they are placed at the original location they occupied before entering the queue. Figure 3 shows workers moving between workstations. The workers that are touching workstations are engaged in the task.

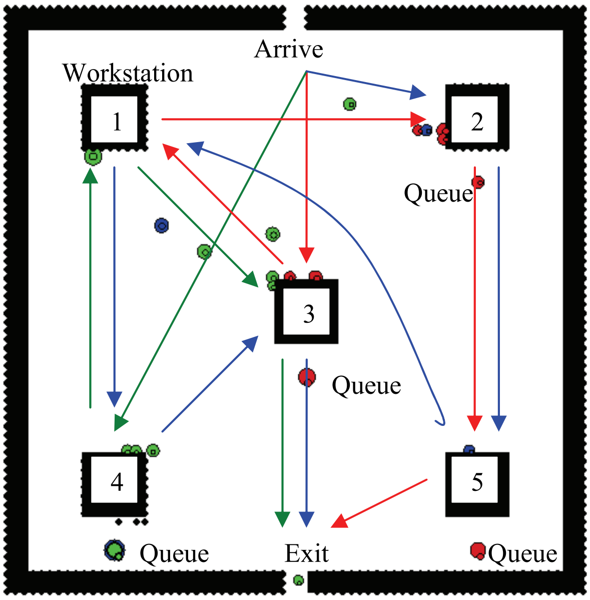


Figure 3: - Simulation Layout: Color Represents Job Type, Circles are Workers, Arrows are Routes

SIMULATION RESULTS

Law and Kelton's discrete-event simulation and our Java based simulation were both run for 98 continuous hours. Tables 3 and 4 show the summary results for Law and Kelton's discrete-event simulation code using the parameter values given in section 4.1.

Table 3: Law and Kelton: By Job Type (hours)

Job Type	Average total delay in queue
1	10.41
2	7.79
3	11.04
Overall average job total delay = 9.23	

Table 4: Law and Kelton: By Workstation

Work station	Ave. # in queue	Average utilization	Ave. delay# in queue (hr)
1	21.11	0.96	5.23
2	8.4	0.96	4.04
3	0.492	0.68	0.13
4	6.56	0.92	2.43
5	0.909	0.75	0.46

Tables 5 and 6 show the summary results for our pedestrian and job shop model. A total of 1397 workers completed their jobs during the 98 hour simulation.

Table 5: Simulation 1: Job Type (hours)

Job Type	Average total delay in queue
1	1.07
2	1.75
3	1.09
Overall average job total delay = 1.27	

Table 6: Simulation 1: By Workstation

Work station	Ave. # in queue	Average utilization	Ave. delay# in queue (hr)
1	6.41	0.91	1.54
2	16.42	0.96	1.05
3	12.81	0.98	1.18
4	7.49	0.94	2.48
5	0.11	0.42	0.06

5. ANALYSIS

The results of the two simulations differ, our pedestrian and job shop simulations have lower overall job and workstation queuing delays. The average numbers of individuals in the queues are within the same range but are ordered differently within the workstations. While one simulation run cannot provide conclusive results the difference may be explained by two factors. First the variability in inter-arrival times and task service times due to the distributions used to obtain there values. Second the inclusion of the explicit worker movement.

The time it takes workers to move between workstations and find an open space so they may be next to the workstation (they only attach to workstation if they occupy a neighboring cell) is not included in their time in queue. Their delay in queue is only calculated as the time workers actually enter the queue until the time they leave the queue. Movement between workstations in Law and Kelton's model is instantaneous thus it should be expected that queue delay times in their simulation would be somewhat higher.

Table 6 gives the minimum distances workers would have to travel along the different job routes. The movement model is probabilistic so workers would rarely if ever take the most direct route. Avoidance maneuvers (avoiding other people and obstacles) also increase the actual traveling distances and times for workers, who move at a velocity of 1.2m/s.

Table 6: Travel Distance by Job (meters)

Job Type	Center to Center Distance
1	59.02
2	51.87
3	81.6

6. SUMMARY

Our pedestrian motion model allows individuals to choose different paths on the fly. This allows for a much greater application base wherein the CA models of pedestrian motion can be used. Dynamic destination choice allows for implementation of greater agent intelligence, giving them the ability to address resource needs and modifying movement to optimize workflows. Job shop workflow models previously done by discrete event simulations did not consider the dynamics of the assembly operator's movements. This technique would allow workflow analysis to be done by not only considering the machinery processes being used, but the environment, the individual to individual interactions, and individual to workstation interactions which take place throughout the process.

ACKNOLWLEDGMENTS

This material is based upon work supported by the National Science Foundation under Grant No. BCS-0527545

REFERENCES

AlGadhi, S.A, Mahmassani, H.S., and Herman, R., 2002. A speed-concentration relation for bi-directional crowd movements with strong interaction. In: M. Schreckenberg and S.D. Sharma, eds. *Pedestrian and Evacuation Dynamics*, Berlin, Germany: Springer-Verlag, 3-20.

Batty, M., DeSyllas, J. and Duxbury, E., 2002. The discrete dynamics of small-scale spatial events: Agent-based models of mobility in carnivals and street parades. *International Journal of Geographical Information Science* 17(7):673-697.

Blue, V.J. and Adler, J.L., 2002. Flow capacities from cellular automata modeling of proportional splits of pedestrians by direction. In: M. Schreckenberg and S.D. Sharma, eds. *Pedestrian and Evacuation Dynamics*, Berlin, Germany: Springer-Verlag, 115-122.

Celano, G., Costa, A., Fichera, S., and Perrone, G., 2004. Human factor policy testing in the sequencing of manual mixed model assembly lines. *Computers & Operations Research* 31(1): 39-59.

Dijkstra, J., Jesurun, J., and Timmermans, H., 2002. A multi-agent cellular automata model of pedestrian movement, In M. Schreckenberg and S.D. Sharma, eds. *Pedestrian and Evacuation Dynamics*, Berlin, Germany: Springer-Verlag. 173-180.

Helbing D. and Molnar, P., 1995. Social force model for pedestrian dynamics. *Phys. Rev E.* 51:4282-4287.

Helbing, D. Farkas, I., and Vicsek, T., 2000. Simulating dynamical features of escape panic, *Nature* 407:487-490.

Hoogendoorn, S.P., Bovy, P.H.L., and Daamen, W. 2002. Microscopic pedestrian wayfinding and dynamics modeling. In: M. Schreckenberg and S.D. Sharma, eds. *Pedestrian and Evacuation Dynamics*, Berlin, Germany: Springer-Verlag, 173-180.

Hughes, R.L., 2002. A continuum theory of pedestrian motion, *Transportation Res. B.* 36:507-535.

Kessel, A., H. Klüpfel, J. Wahle, and M. Schreckenberg. 2002. Microscopic simulation of pedestrian crowd motion. In M. Schreckenberg and S.D. Sharma, eds. *Pedestrian and Evacuation Dynamics*, Berlin, Germany: Springer-Verlag. 193-202.

Law, A.M. and Kelton, D.M., 2000. *Simulation Modeling and Analysis 3rd ed.* Boston: McGraw-Hill Higher Education.

Luke, S., Cioffi-Revilla, C., Panait, L., Sullivan, K., and Balan, G., 2005. MASON: A Multiagent Simulation Environment. *Simulation* 81(7):517-527.

MacCarthy, B., Wilson, J., and Crawford S., 2001. Human Performance in Industrial Scheduling: A Framework for Understanding. *Human Factors and Ergonomics in Manufacturing* 11(4):299-320.

Schadschneider, A. 2002. Cellular automaton approach to pedestrian dynamics – theory. In: M. Schreckenberg and S.D. Sharma, eds. *Pedestrian and Evacuation Dynamics*, Berlin, Germany: Springer-Verlag. 76-85.

Wang, Q., Owen, G.W. and Mileham, A.R., 2005. Comparison between fixed- and walking-worker assembly lines. *Proceedings of the I MECH E Part B Journal of Engineering Manufacture* 219(4): 845-848.

AUTHORS BIOGRAPHY

Alan Jolly graduated from the University of South Alabama with a B.S. in physical geography. He obtained his M.S. in geography from the University of Tennessee specializing in watershed processes and geographic information sciences. He participated in the post-masters research program at Oak Ridge National Laboratory working on population models and traffic routing algorithms. Currently he is pursing his PhD in Modeling and Simulation at the University of Central Florida under Dr. Kaup.

Rex Oleson II graduated from Susquehanna University with a B.S. in physics and mathematics. Following 4 years of software development, he obtained an M.S. in Mathematics from UCF. He is currently working on a Ph.D in Modeling and Simulation. He works at IST under Dr. Kaup working on pedestrian movement and simulation techniques.

David J. Kaup is Provost Distinguished Research Professor in mathematics at the University of Central Florida with a joint appointment at the Institute for Simulation and Training. He has a PhD in Physics from the University of Maryland. He was a Joint Professor of Physics and Mathematics at Clarkson University before joining UCF. His research interests are in the area of non-linear waves, in particular solitons, and he has been recently applying this background in the area of modeling the dynamics of crowds.

SIMULATION MODEL OF AIRPORT RUNWAY INCURSIONS

Luigi Careddu[(a)], Francesco Costantino[(b)], Giulio Di Gravio[(c)]

[(a)]Alitalia s.p.a.
[(b)]Department of Mechanics and Aeronautics – University of Rome “La Sapienza”
[(c)]Department of Mechanics and Aeronautics – University of Rome “La Sapienza”

[(a)]luigi.careddu@gmail.com, [(b)]francesco.costantino@uniroma1.it, [(c)]giulio.digravio@uniroma1.it

ABSTRACT

Runway incursions are defined by ICAO as “events that create an incorrect presence of aircrafts, vehicles or persons in airport restricted areas of take-off and landing”. In the Italian National Agency for Flight Safety Report (2001) it can be read: “Runway incursions are defined as one of the highest actual risks of aerial transportation in many airports all over the world. The situation is no more acceptable and it’s necessary to apply all the possible countermeasures with the commitment of public and private institutions and operators to solve the problem”. In fact, even if the most serious accidents are very rare (with a probability of about $1/10^7$ movements), their consequences can be catastrophic. The target of the research is to build a simulation model of ground circulation of aircrafts in aerodromes to evaluate frequencies of occurrence of incursions and the connection of the events to set an analysis on parameters as visibility, technological infrastructures and tower controls. The airport “G.B. Pastine” of Ciampino is presented as a case study, identifying actual status and opportune evolutions.

Keywords: runway incursions, airport safety, object-oriented simulation

1. INTRODUCTION

Accidents occurred in Tenerife, 27th March 1977 (583 deaths), and Milano Linate, 8th October 2001 (118 deaths) tragically underlined how safety of passengers and flights is strictly related to the conditions of runways and to the coordination of ground operations.

In the last years, many different studies (Eddowes, Hancox and Mac Innes 2001; Hillestad et al. 1993) estimated the distribution of risk incidents in the different stages of the flight, from take-off to landing. Among these researches, only the most recent reports of Eurocontrol (2004) and Federal Aviation Administration (FAA, 2006) recognized runway incursions as one of the most effective factors that can influence safety: their values are increasing to one incursion a day so to place in “top priority” the reduction and protection of the related accidents.

According to the International Civil Aviation Organization (2004) it’s possible to classify:

- *runway incursion*: any occurrence at an aerodrome involving the incorrect presence of an aircraft, vehicle or person on the protected area of a surface designated for the landing and take-off of aircrafts;
- *accident*: any event associated with the movement of an aircraft, from boarding to landing, where a person is seriously wounded or dead [...] or when the aircraft is structurally damaged [...], missing or completely inaccessible;
- *incident*: any event, not being an accident, associated with the movement of an aircraft that harms or can harm the safety of the flight.

To give an indication of the residual margin of safety, FAA set four degrees of gravity (from higher “A” to lower “D”, depending on the probability of accident and distance of the agents) and three typology of runway incursions, according to the root cause: air traffic control error, pilot error or third party error interfering with the operations. In the cited reports, segmented studies on different aviations and frequency of occurrence in USA, Europe and Italy show the great impact of the phenomena.

Since April 2004, the Provisional Council definitively approved the European Action Plan for the Prevention of Runway Incursions and is now continuing its works to spread and awaken the different players in applying the recommendations of the document, sharing information and introducing standards of data collection and analysis. According to FAA Flight Plan 2005-2009 (2004), all the organizations involved are setting as the main target to reach before 2009 the reduction of runway incursions, through three different strategies:

- identification, reduction and protection of collision risks;
- development of appropriate innovative infrastructures;
- use of advanced simulation models to design and develop new equipments, procedures and training.

2. THE SIMULATION MODEL

An airport is an extremely complex system where thousands of peoples and many different operations are arranged according to accurate procedures.

The target of this study consists of creating a model of airport ground operations (in particular aircraft movements) to carry out a risk analysis of runway incursions, investigating in details all the possible fault events and consequences that can create incidents or accidents. To this extent, considering Wyss, Craft and Funkhouser (1999), an object-oriented analysis was implemented, defining:

- entities with attributes to describe their custom characteristics;
- states to draw their interaction with internal and external environment.

Objects, connected in subsystems and systems, are parametrically modelled in C++ and communicate through messages to represent information, materials or energy flows. Furthermore, an extraction of pseudo-random numbers, associated to normal distribution of probability for event's occurrence, allows an automatic generation of scenarios as:

- for each object, different behaviours are simulated, both correct and erroneous (i.e. an aircraft at a stop bar connection, without clearance, can respect the signal or enter the runway);
- the combinations of objects and events represent all the possible alternatives comprehensive of their different degree of risks, according to FAA classification;
- the evolution of systems and subsystems can identify the most frequent dynamics and their influent parameters.

As shown in figure 1, the model represents the interaction, in terms of runway incursions, of three type of elements (the airport, the aircrafts and the control tower) that act independently and are coordinated by a communication network with a black box to record any evolution of the whole system.

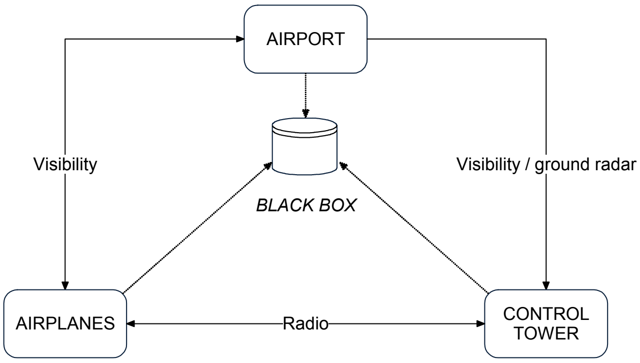


Figure 1: Communication framework

2.1. Airport model element

The object represents the environment where the aircrafts and the tower interact. It's constituted of two main elements:

- *the airport map:* it represents the physical set and geometry of all the runways, taxiways, aprons and paths, opportunely divided into coded segments. These information are transmitted to any aircraft, simulating the perception of the pilot on the external environment and his possibility to consult the map or read the signals along the routes;
- *the aircraft list:* it consists of the registration of the correct position of any aircraft in the different segments with its destination and orientation. For example, it allows the pilots to locate, real-time, aircraft taxing position, preserve an opportune distance, verify if a runway is free and monitor any eventual incursion.

To simulate incidents and accidents, causes of errors from the environment were modelled considering:

- unauthorized planes that cross the runways or taxiways;
- authorized planes with an *"hold short of the runway"* command confirmed but bypassed.

2.2. Aircrafts model element

Each aircraft is identified by a set of parameters to define both physical characteristics (speed, acceleration, position, dimensions) and information (visual or by radio) that can evolve, during the simulation and according to the other elements of the models, through "flying", "moving" and "waiting" states. Two lists manage the dynamics of any aircraft:

- a waypoint list containing all its expected positions;
- a to-do list with all the tasks to be executed.

The tasks-cycle of a generic aircraft is represented in figure 2. To simulate incidents and accidents, five causes of errors due to pilots were modelled considering:

1. mistakes on *"read-back"* or *"hear-back"* orders assigned;
2. correct *"read-back"* or *"hear-back"* but different tasks executed;
3. no respect of the command *"hold short of the runway"* and cross the runway or stop taxing on active runway;
4. misunderstanding of radio communication addressed to other pilots;
5. different levels of reactivity.

In table 1, the different human error causal factors are classified according to Marguglio's framework:

- knowledge-based: lack of knowledge of the standard, requirement or need;
- cognition-based: lack of the appropriate level of cognition; lack of ability to understand, apply, analyze, synthesize or evaluate such as to be able to meet the standard, requirement or need;
- value-based or belief-based: lack of respect for or acceptance of the standard, requirement or need;
- error-inducing condition-based or error-likely situation-based: lack of recognition of the condition or situation and/or lack of counteracting behaviour;
- reflexive-based: lack of thought processes and behavioural techniques for conservative decision-making in reacting to an immediate "field stimulus";
- skill-based: lack of dexterity;
- lapse-based: nothing lacking; simply “blew it”.

Table 1: Classification of defined pilot errors

	HUMAN ERRORS				
CAUSAL FACTOR	***1***	***2***	***3***	***4***	***5***
Knowledge-based	🟨				
Cognition-based	🟨			🟨	
Value/Belief-based	🟨	🟨	🟨		
Error-Inducing Condition-based	🟨			🟨	
Reflexive-based	🟨				🟨
Skill-based					🟨
Lapse-based	🟨	🟨	🟨	🟨	🟨

2.3. Control tower model element

The object models both the behaviour of the control tower and ground control where the different tasks are divided into two categories: management of clearances and monitor of the ways. The first consists of:

- requests with instantaneous responses, for communications or information the aircraft should know while approaching (i.e. its landing runway, already communicated by the Approach Control);
- requests of path, to indicate the correct route to use to reach a determined point, with a FIFO logic and different level of priority (communications with the Ground Control);
- requests of clearance, with an order list subjected to strict procedures (radio traffic with the Control Tower).

Furthermore, the tower monitor ways and corridors with a frequency depending on the reaction of the operators to avoid eventual faults. When the tower identifies an incorrect action of an aircraft (ex. an alignment for take-off without clearance), it analyzes the situation, verifying the presence of other aircrafts and their relative positions, and gives orders trying to avoid the collision.

To simulate incidents and accidents, causes of errors made by control tower were modelled considering:

- a controller forgetting an airplane, a vehicle, a given clearance, the runway or taxiway state;
- communication errors like misunderstandings or mistake in read-back;
- a controller incorrectly estimating relative distances of aircrafts.

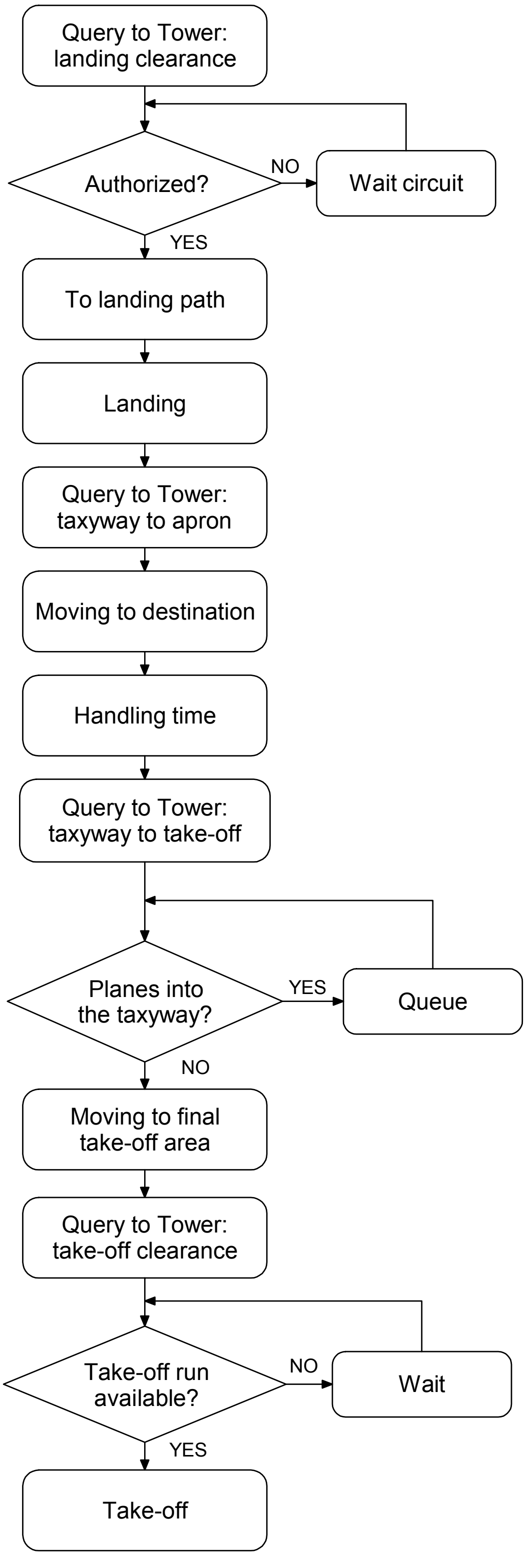

Query to Tower: landing clearance
Authorized?
NO
Wait circuit
YES
To landing path
Landing
Query to Tower: taxyway to apron
Moving to destination
Handling time
Query to Tower: taxyway to take-off
Planes into the taxyway?
YES
Queue
NO
Moving to final take-off area
Query to Tower: take-off clearance
Take-off run available?
NO
Wait
YES
Take-off

Figure 2: Life-cycle of aircraft element

2.4. Communication network model element

Communications act in two different ways: radio Ground/Air/Ground and visual perception of objects in the environment. Any aircraft interfaces the tower by radio communication to make requests and receive clearance, but instructions are subjected to delays, reaction times and human errors (wrong communication or wrong destination).

The complete network allows the model to know the correct position of all the aircrafts in the airport: each object can so visually identify its relative position while the tower can interrogate the airport to know the position of the aircrafts on the ground.

2.5. Black box model element

To gather information on the evolution of the model, a black box was modelled to record various information coming from the objects, to describe not only dynamics of the incursions but the complete sequence of events, states, properties and communications. It can be represented as a buffer of information, with a structured output including the list of all the faults, a summary of the simulation and a detailed log for the analysis of the events.

3. THE CASE STUDY

3.1. Ciampino airport

The simulation model was applied to the Ciampino International Airport of Rome, a medium size structure, with a traffic of about 30.000 movements/year and a positive trend of 30% in the last years, both on military (18%) and civil aviation (82%) .

The schematic layout, shown in figure 3, is very simple: a single runaway, with two threshold (15 and 33) and two parallel taxiways, A on the south side and B on the north side, currently unusable. Taxiway A takes to the main apron (south) and to a secondary apron (north) actually dedicated to military traffic. All the procedures are so simulated with compulsory routes and task lists coherent to the layout.

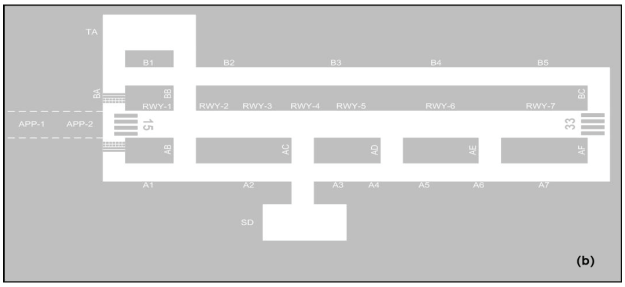


Figure 3: Layout of Ciampino Airport

Civil aviation uses threshold 15 as preferential and 33 only seldom, daily and with optimal visibility. Standard procedure commands to free the runway on the right, using the first possible junction, except for commercial aviation (mainly B737) that uses typically the end of runway junctions; subsequently the aircrafts exit on the left to the southern apron. Military flights, destined to northern apron, approach threshold 15 and, being taxiway B not accessible, have to free on the left, cover all the taxiway, get the junction and cross again the runway before destination.

ANSV data (2002, 2003 and 2004), direct interviews with airport personnel and comparison with similar structures (for visibility and traffic volumes) showed an indication of about 5 incursions every 100.000 movements for a total of 15 in about 10 standard years. As the only root cause of incursions is an unauthorized entry for take-off, the probability of wrong occupation is about $1/10^4$.

As ICAO doesn't give an analytic specification on how to classify incursions, it's first necessary to define these rules depending on the presence of vectors on runways and thresholds, their relative position and speed. In the model it is assumed that, when an aircraft is entering the runway without authorization, the incursion will be classified as follows:

- *class A:* another aircraft on the runway with relative speed higher than 70 m/s;
- *class A:* no aircraft on the runway but at least one approaching the threshold from a distance of less than 1 km;
- *class B:* another aircraft on the runway with relative speed lower than 70 m/s;
- *class C:* no other aircraft on the runway but at least one approaching the threshold from a distance of more than 1 km;
- *class D:* any other case configuring an incursion.

3.2. Standard conditions

A sensitivity analysis on 6.000.000 movements with an average traffic of 27.000 (correspondent to year 2002 data) can show the variation of A+B incursions every 100.000 movements (FAA standard indicator), depending on the system main parameters.

The control time Δt_c (figure 4) of the tower models the level of attention of human and technological support infrastructures. This points out as an interval of about 5 seconds (ex. granted by an Advance Surface Movement and Guidance Control System A-SMGCS) allows a significant reduction of the parameter, while a reduction of the frequency generates a subsequent sharp increase that tends to an asymptote when control interventions become ineffective.

Once defined the standard level of the two parameters, the simulation returns a global rate of incursions of 5,37 every 100.000 movements with 1,06% of A+B for a total of 0,057 (every 100.000 movements).

The visibility harshly affects the number of incursions when decreasing below 1300m (the critical distance from the control tower to the threshold 15 and its two junctions AA and AB), as shown in figure 5. In these conditions, both pilots and control tower reduce their ability to relieve incursions, perceive dangerous situations and carry out corrective actions.

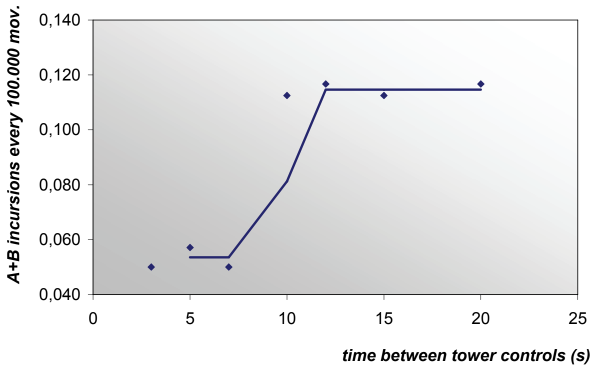


Figure 4: A+B incursions depending on control interval.

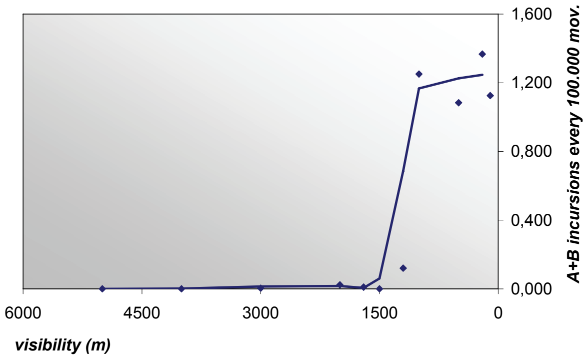


Figure 5: A+B incursions depending on visibility.

3.3. Airport evolution

Considering the fast development of Ciampino in the last years, mainly due to low-cost flight companies, it's necessary to show the relative impact it had. The target of the simulation is to make a further analysis of A+B incursions, starting from the actual traffic to evaluate the capacity limit of the airport and possible evolutionary scenarios.

Structural interventions, already planned, define a new configuration where 40% of the traffic is destined to north apron (opening it to civil aviation) and a potential restoration of the taxiway B, to simplify the circulation (figure 6).

Figure 7 shows the general trend of the two configurations to notice, as expected, a direct increase in the number of incursion with the airport congestion. Considering a traffic of 36.000 movements/year that soon will involve Ciampino, the evolved situation allows a risk reduction of about 50% and a diminution of total incursions rate from 6 to 5,1. Furthermore, the restoration of taxiway B generates an increase of airport capacity, contemporarily diminishing the risk of accidents in accordance to FAA standard requirements. In fact, the actual ratio of 0,057 can be further reduced of 15% in four years, granting a further residual capacity.

4. CONCLUSIONS

The final results of the simulation consist of an evaluation of runway incursions risk level of a particular airport, segmented in classes of gravity.

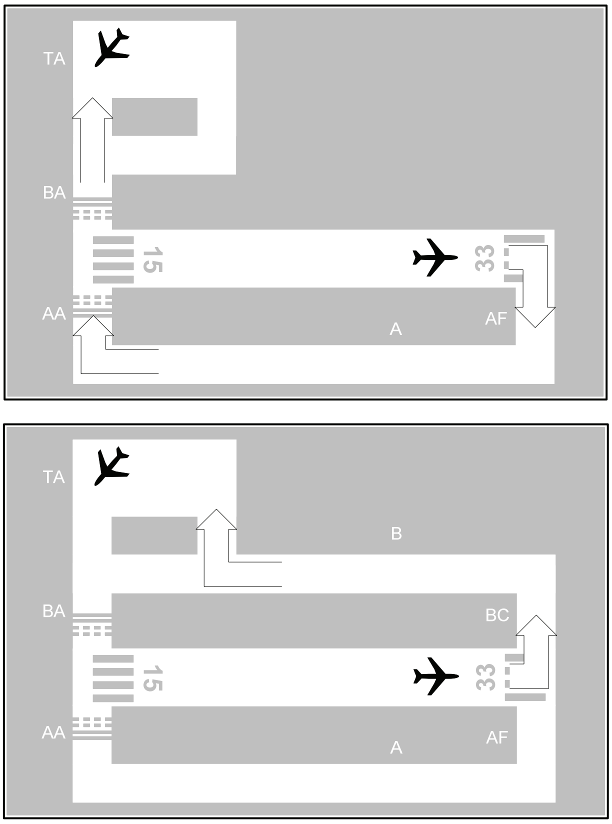


Figure 6: Layout evolution.

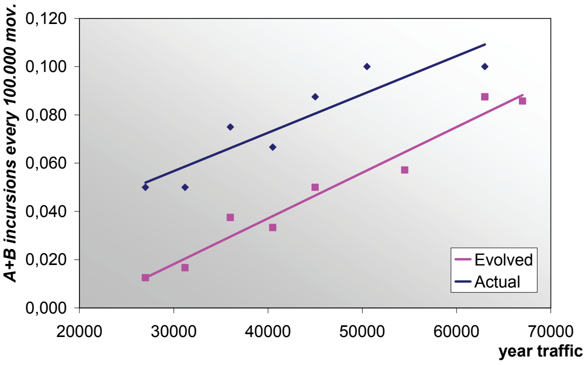


Figure 7: A+B incursions in layout evolution.

The study is in still its first stages of evolution and particular refinements can be applied. First of all, a simple integration of vehicles and pedestrians could complete the possible root causes. Furthermore, a collection and acquisition of specific data (ex. detailed analysis of the airport signs, impact on pilot's perception and integration of studies on human reliability) will tune the model so to strictly define lines of intervention and priority, both technical and organizational.

The object-oriented technique has shown, even in a feasibility and demonstrative application, that the approach allows to simulate a complex environment without defining a priori scenarios and dedicated analysis, as for fault-tree or event-tree, only observing the evolution of the system.

Sensibility analysis can be easily implemented to evaluate the impact of the structural parameters, the definition or re-arrangement of circulation procedures of aircrafts and adoption of innovative informative, visual or control tools. All the investments, on people, infrastructures and management logic, can so be tested and classified in terms of protection or reaction to events, defying the most opportune level of safety to characterize the airport.

In the end, the model has to be supported by an accurate cost-benefit analysis to list and compare opportunities and criticalities. Once this evolution is completed, the model can be simply added as an independent engine in all the applications that simulate standard operations of an airport (both commercial or dedicated), introducing systematic levels of uncertainty and alarms to define, implement or set effectiveness of recovery actions.

REFERENCES

Adelantado M., 2004. *Rapid Prototyping of Airport Advanced Operational Systems and Procedures through Distributed Simulation*, Simulation, 80 (1), 5-20.

Agenzia Nazionale per la Sicurezza del Volo (Italian Agency for Flight Safety), 2004. *Rapporto Informativo sull'Attività Svolta dall'Agenzia* (Annual report).

Agenzia Nazionale per la Sicurezza del Volo (Italian Agency for Flight Safety), 2003. *Rapporto Informativo sull'Attività Svolta dall'Agenzia* (Annual report).

Agenzia Nazionale per la Sicurezza del Volo (Italian Agency for Flight Safety), 2002. *Rapporto Informativo sull'Attività Svolta dall'Agenzia*. (Annual report).

Agenzia Nazionale per la Sicurezza del Volo (Italian Agency for Flight Safety), 2004. *Relazione d'Inchiesta (Inquiry Report) - Incidente Occorso agli Aeromobili BOEING MD-87, Marche SE-DMA e CESSNA 525-A, Marche D-IEVX, Aeroporto Milano Linate, 8 ottobre 2001.*

Eddowes M., Hancox J., Mac Innes A., 2001. *Final Report of the Risk Analysis in Support of Aerodrome Design Rules.* Norwegian Civil Aviation Authority.

Ente Nazionale per l'Aviazione Civile (Italian Authority for Civil Aviation), Servizio Studi e Programmazione, Ufficio Studi e Statistiche, 2002. *Annuario Statistico* (Annual statistical report).

Ente Nazionale per l'Aviazione Civile (Italian Authority for Civil Aviation), Servizio Studi e Programmazione, Ufficio Studi e Statistiche, 2003. *Annuario Statistico* (Annual statistical report).

Eurocontrol, Safety Regulation Commission (SRC) 2004. *Annual Safety Report.*

Eurocontrol, 2004. *European Action Plan for the Prevention of Runway Incursion.*

Hillestad R. et al., 1993. *Airport Growth and Safety - A Study of the External Risks of Schiphol Airport and Possible Safety-Enhancement Measures,* RAND.

Federal Aviation Administration, 2006. *Runway Safety Report: Runway Incursion Trends and Initiatives at Towered Airports.*

Federal Aviation Administration, 2004. *Flight Plan 2005-2009.*

Federal Aviation Administration, 2004. *Flight Plan 2004-2008. in the United States, FY 2000 - FY 2003.*

International Civil Aviation Organization, 2004. *Advanced Surface Movement Guidance and Control System Manual*, 1st Edition.

International Civil Aviation Organization, 2001. *Procedures for Air Navigation Services - Air Traffic Management. Doc. 4444/ATM 501.* 14th Edition.

International Civil Aviation Organization, 2001. *Annex 14 to the Convention on International Civil Aviation*, 9th Edition.

Janic M., 2000. *An Assessment of Risk and Safety in Civil Aviation.* Journal of Air Transport Management, 6, 43-50.

Wyss G.D., Craft R.L., Funkhouser D.L., 1999. *The Use of Object-Oriented Analysis Methods in Surety Analysis.* Sandia National Laboratories, Albuquerque, NM.

AUTHORS BIOGRAPHY

LUIGI CAREDDU

BSc in Aerospace Engineering, he is a member of the Extended Range Operations - Reliability Program of Alitalia.

FRANCESCO COSTANTINO

BSc in Mechanical Engineering, Master Degree in Quality Management and Engineering and Ph.D. in Engineering of Industrial Production, he is an assistant professor of Mechanical and Industrial Plants at the Faculty of Engineering, University of Rome "La Sapienza".

GIULIO DI GRAVIO

BSc in Mechanical Engineering, Master Degree in Quality Management and Engineering and Ph.D. in Engineering of Industrial Production, he is a researcher of Mechanical and Industrial Plants at the Faculty of Engineering, University of Rome "La Sapienza".

MATHEMATICAL MODELS OF GAS FIRED BOILERS

Renzo Tosato[(a)]

[(a)]Department of Mechanical Engineering, University of Padova, Padova, Italy,

[(a)]renzo.tosato@unipd.it

ABSTRACT

As for traditional boilers, the cyclical efficiency of a condensing boiler can be defined by its experimental efficiency at full load, for a given return water temperature and by its experimental stand-by losses, considering not only the boiler but its actual regulation system too. Measurements of natural gas consumptions during 8 years will be presented and used to simulate conventional and condensing boilers in domestic heating plants. To calculate seasonal efficiency of a boiler, a sinusoidal variation of loads during the season, from zero to a maximum value, can be assumed.

Keywords: gas, boiler, condensing, efficiency

1. NOMENCLATURE

AREA	surface of flat	$[m^2]$
days	= 1, 2, .. day0 during a season	[-]
days0	total days in a season	[-]
GAS	gas consumption during time	$[m^3]$
GAScons	gas from first day	$[m^3]$
GASy	total gas in a heating season	$[m^3]$
H_i	high heating value of the fuel	$[J/ m^3]$
LU	load	[-]
LUmax	maximum load	[-]
m	mean seasonal load	[-]
P_{cons}	input of the boiler	[kW]
P_{fl}	flue gas losses at partial load	[kW]
P_{fl0}	flue gas losses at full load	[kW]
PN	nominal output	[kW]
PR	experimental stand-by losses	[kW]
PR_X	stand-by losses at null load	[kW]
P_{sl}	surface losses at partial load	[kW]
PU	partial output of the boiler	[kW]
P_{vl}	ventilation losses at partial load	[kW]
P_{vl0}	ventilation losses at null load	[kW]
T	time-step	[hour]
TM	outlet water temperature	[°C]
TR	return water temperature	[°C]
TR*	nominal return water temp.	[°C]
TRp	design water temperature	[°C]
TR_y	water temperature at full load	[°C]
η^*	nominal efficiency at full load	[-]
η_0	experimental efficiency at full load	[-]
η_b	combustion efficiency	[-]
η_c	cyclical efficiency at partial load	[-]
η_{dc}	efficiency of distribution system	[-]
η_{ec}	cyclical efficiency of emitters	[-]
η_g	seasonal efficiency	[-]
η_{gc}	efficiency of boiler and regul. syst,	[-]

2. INTRODUCTION

Approximately 40% of final energy is consumed in buildings in the whole European Union (Eurostat 2007). Much attention is paid to the improved energy efficiency in building sector during the last years, because the sector harbours a considerable potential of primary energy saving and reduction of emissions, having a negative impact on the environment.

A large proportion of the European old housing stock consists of family houses, very often with a central boiler and no temperature control at the individual apartment level. Very often refurbishment takes place, involving replacement of the boiler (usually gas), incorporating central thermostatic control or thermostatic valves, while the pipes and radiators remain the same. It is very difficult to predict fuel consumption, control behaviour of the system and quality of the indoor environment (temperature amplitudes, etc.).

In order to build energy new efficient buildings, correct decisions have to be made already at the design stage. Heating system is among the systems, which are responsible for the consumption of the major part of energy in cold climate countries. It means that the design of this system can influence the overall energy performance of the building.

The system designer faces the challenge to select a suitable boiler in terms of lightweight or cast-iron, condensing or non-condensing, originally specified capacity (i.e. oversized) or re-calculated, etc. Some of the factors that should be considered include: boiler type, design temperatures, capacity of the system, control of the system (local, central, or both), thermal inertia of building and heating plant.

Requirements to evaluate the impact on environment of the energy-using products during its whole life cycle are set in directive 2005/32/EC of the European Parliament and of the Council of 6 July 2005 establishing a framework for the setting of eco-design requirements for energy-using products and amending Council Directive 92/42/EEC and Directives 96/57/EC and 2000/55/EC of the European Parliament and of the Council. Directive 2006/32/EC of the European

Parliament and of the Council of 5 April 2006 on energy end-use efficiency and energy services and repealing Council Directive 93/76/EEC recommends to use minimised life cycle cost analysis in particular cases (European Commission 1997).

A number of studies about and energy systems of building life cycle have been conducted. Some of them deal with the building construction and energy systems' life cycle. Some of them analyse energy systems (Prek 2004).

Analysis shows that in all cases the operation stage is most important, and there the biggest energy consumption occurs. That means that it is not so important in this case what materials or elements are used in the system, much more important is the efficiency of the heat generator.

The processes occurring in apartment heating can be analyzed and simulated by computer programs. These programs examine various situations in order to evaluate heat losses and seasonal fuel consumption.

Every gas-heating system includes a gas-fired boiler, a distribution system, a set of emitters and a control device of temperatures in the apartment. During a time-step *T,* the energy balance between energy contributions from emitters and the fuel consumption GAS is:

$$\text{Energy from heat emitters} = \text{GAS}\bullet H_i \bullet \eta_c \qquad (1)$$

where H_i is the high heating value of the gas, and η_c is the average efficiency of the boiler and the distribution system during the same time-step *T*.

These so-called cyclical efficiency η_c may be considered the product of the three cyclical efficiencies which may be attributed to the three subsystems in which the heating plant can be divided (Rosa and Tosato 1985):

$$\eta_c = \eta_{gc} \bullet \eta_{ec} \bullet \eta_{dc} \qquad (2)$$

where η_{gc} = cyclical efficiency of the boiler and its regulation system; η_{dc}=cyclical efficiency of the distribution system; η_{ec}=cyclical efficiency of the emitters.

The boiler (conventional and condensing) is taken into account ($\eta_{dc} = \eta_{ec} = 1$) in this paper. However, as the regulation system affects the temperatures of the boiler and its efficiency, it is more correct to examine a boiler together with its regulation system η_{gc} instead of the boiler alone.

3. CONDENSING BOILER

The main differences between conventional systems and condensing boilers are:

- Condensing boiler technology requires that the exhaust gases are removed with *fans*, since, because of the strong cooling of the exhaust gases, their buoyancy is generally not sufficient.
- The exhaust gas pipes have to be *insensitive to humidity* because exhaust gases contain rests of humidity which condense in the exhaust gas pipes.
- *Condensate* has to be removed.
- The parts of condensing boilers, of the exhaust gas removal system and of the condensate removal system have to be *corrosion* resistant.
- The *second heat exchanger* was one of the first special constructional features in the beginning of the condensing boiler technology development. In order to use the heat energy of the exhaust gases an additional heat exchanger was added to a conventional boiler. Today such a heat exchanger is still used for large high-performance boilers.

4. MODEL OF BOILER OPERATING AT PARTIAL LOAD

There are numerous energy losses from a boiler. While most of them are minor, and comprise about 1% or less of fuel input, there are two or three major losses that typically represent 10 to 20 % of fuel input.

As described by Natural Resources Canada's Office of Energy Efficiency (OEE) (2006), Flue gas losses represent the heat in the flue gas that is lost to the atmosphere upon entering the stack. Stack losses depend on fuel composition, firing conditions and flue gas temperature. They are the total of two types of losses:

- Dry Flue Gas Losses: the (sensible) heat energy in the flue gas due to the flue gas temperature;
- Flue Gas Loss due to Moisture – the (latent) energy in the steam in the flue gas stream due to the water produced by the combustion reaction being vaporized from the high flue gas temperature.

Radiation and convection losses are independent of the fuel being fired in a boiler and represent heat lost to the surroundings from the warm surfaces of a high-temperature water generator. These losses depend mainly on the size of the equipment (e.g., small boilers have a proportionately larger percentage loss than large boilers), and the actual output relative to the maximum design output.

Considering a model 1, the major energy losses associated with boilers fall into three categories: flue gas losses to the chimney Pfl during the ON time, radiation and convection losses of external surfaces Psl during the ON and OFF times, ventilation losses to the chimney during the OFF time.

For condensing boilers it is very difficult to evaluate these losses by experimental tests at full load, at partial loads and at null load.

I prefer to consider for condensing and traditional boilers a different model (Model 2), more near to the experimental possibilities in a test rig (Rosa and Tosato 1988) .

A linear relationship between energy consumed and load, and steady-state model were considered as appropriate to calculate the energy consumption over a long period as during the winter period.

This linear law, referred to a unit of time and written in an adimentional form, is:

$$P_{cons}/PN = A + B \bullet PU/PN \qquad (3)$$

where P_{cons} = input of the boiler; PN = nominal output of the boiler; PU = partial output of the boiler during the operation; A, B = two constants.

This law is acceptable if the mean temperature of the water in the boiler is kept constant and is more or less valid for traditional boilers connected to a plant by a control mixing valve. As far as condensing boilers are considered, the temperature of the return water varies and a linear relationship similar to eqn. (3) is acceptable only for a narrow range of loads. For the whole load range the variation of flue gas losses P_{fl}, ventilation losses P_{vl} and surface losses P_{sl} have to be taken into account.

The author analyses conventional boilers connected to a plant with constant temperatures of water when the load varies. For this type of boilers he suggests that equation 4 should be used, where, differently from equation 3, the functions A and B (eq. 5 and 6) are not constant . Losses P_{fl}, P_{vl} and P_{sl} depend by a linear equation on loads and can be assessed by experimental tests at full load (i.e. P_{fl0}=0.2 and P_{sl0}=0.1) and at null load (i.e. P_{vl0}=0.3). The results of this simulation (Model 1) are plotted in fig. 1.

A second model (Model 2) of losses in a boiler is more near to experimental tests. The major energy losses associated with boilers fall into two categories:

- flue gas losses to the chimney during the ON time, radiation and convection losses of external surfaces Psl during the ON and OFF times, ventilation losses to the chimney during the OFF time.

As indicated in [2], a numerical model of a boiler can be defined by its experimental efficiency at full load η_0, at different return water temperatures TR, and by its experimental stand-by losses PR. It can be assumed that the operation of a boiler with an ON-OFF cyclical mode is based on the following assumptions:

- during the ON-time, the flue gas losses and the surface losses, as well as the efficiency, are the same as during the full-load operation;
- during the OFF-time, the efficiency is zero, and the ventilation and surface losses are a function of the mean temperature (TR + TM)/2 of the water, where TM is the outlet water temperature.

For condensing boilers and for conventional boilers too, the author considers stand-by losses PR at null load and steady efficiency η_0 at full load with constant temperatures of water when the load varies (Model 2). He suggests for them that equation 7 should be used, where the functions A and B (eq. 9 and 10) are not constants. Losses PR can be assessed by experimental tests as a function of mean water temperature in the boiler at null load. The results of this simulation are plotted in fig. 2 for condensing boilers (Model 2) and in fig.3 for traditional boilers (Model 2). The results of fig. 3 are the same of fig. 1.

The results of this simulation for condensing boilers with negligible stand-by losses PR and return water temperature TR proportional to the load are plotted in fig. 4 (Model 2).

If the boiler temperature (TR + TM)/2 is taken as constant and the losses $P_{vl} = P_{vl0} \bullet (1 - PU/PN)$ and $P_{fl} = P_{fl0} \bullet PU/PN$ are proportional to PU, according to Fig. 1, then:

$$\begin{aligned} P_{cons}/PN &= 1/\eta_c \bullet PU/PN \qquad (4)\\ &= P_{fl}/PN + P_{vl}/PN + P_{sl}/PN + PU/PN\\ &= P_{fl0}/PN \bullet PU/PN + P_{vl0}/PN \bullet (1-PU/PN) +\\ &\quad + P_{sl0}/PN + PU/PN\\ &= (1/\eta_b - 1) \bullet (1+P_{sl0}/PN) \bullet PU/PN + P_{vl0}/PN \bullet (1-PU/PN) +\\ &\quad + P_{sl0}/PN + PU/PN\\ &= A + B \bullet PU/PN \end{aligned}$$

where η_b = combustion efficiency, and

$$A = P_{vl0}/PN + P_{sl0}/PN \qquad (5)$$

$$\begin{aligned} B &= 1 + P_{fl0}/PN - P_{vl0}/PN \qquad (6)\\ &= 1 + (1/\eta_b - 1) \bullet (1+P_{sl0}/PN) - P_{vl0}/PN \end{aligned}$$

The experimental evaluation of η_b of traditional boilers doesn't present any problem and the Siegert formula ($\eta_b = 1 - C \bullet (T_f - T_a)/CO_2$) is generally used in laboratory and field tests. However, this evaluation doesn't take into account the condensing possibilities and refers to the low heat value. The calculation of P_{sl0} and P_{vl0} losses needs a monitoring rig usually employed only in a laboratory. These difficulties lead to the choice of another equation of energy balance

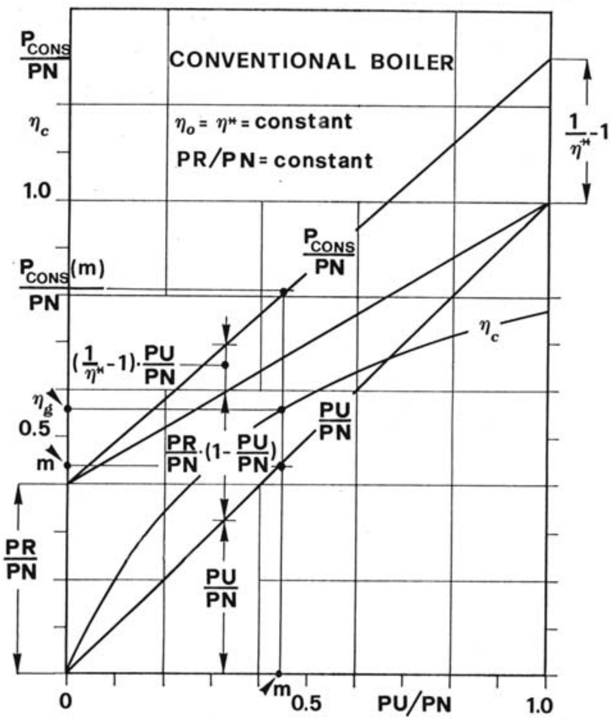


Figure 1. Acceptable model of a boiler at partial load when temperatures are more or less constant, using P_{fl}, P_{vl} and P_{sl} losses. (Model 1)

Accordig to Fig. 2, then it follows that:

$$
\begin{aligned}
P_{cons}/PN &= 1/\eta_c \bullet PU/PN \qquad (7)\\
&= PR/PN \bullet (1-\eta^*/\eta_0 \bullet PU/PN)\\
&\quad + (1/\eta_0 - 1) \bullet PU/PN + PU/PN\\
&= PU/(PN \bullet \eta_0) + PR/PN \bullet (1-\eta^*/\eta_0 \bullet PU/PN)
\end{aligned}
$$

where η^* is the nominal efficiency at full load, corresponding to the nominal return water temperature TR* (usually 60 °C).

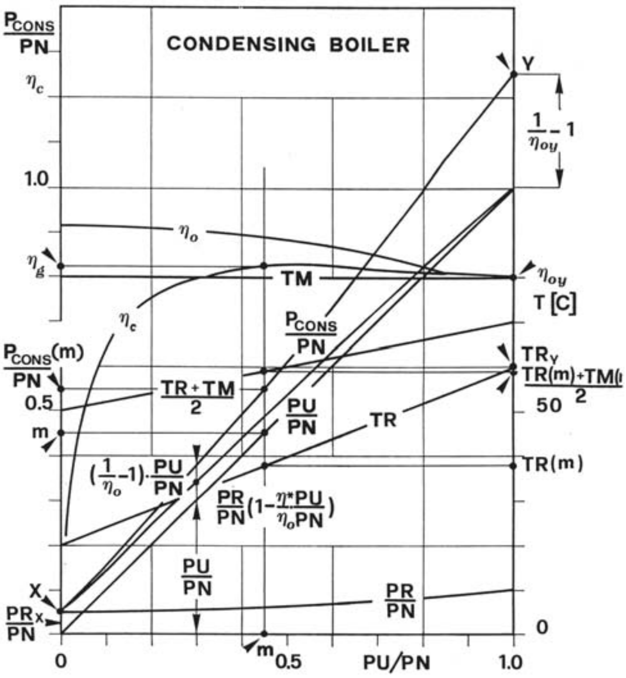


Figure 2. Model of boiler at partial load, using stand-by losses and efficiency η_0 at full load, considering their variation with water temperatures. Condensing boiler connected to plant/system by a 3-way mixing valve.

As represented in Fig. 3, if $\eta_0 = \eta^*$, η_0 and PR are constant (as in a boiler at constant water temperature), the linear relationship, eqn. (4), for the cyclical efficiency, is:

$$
\begin{aligned}
P_{cons}/PN &= 1/\eta_c \bullet PU/PN\\
&= PR/PN + (1/\eta^* - PR/PN) \bullet PU/PN\\
&= A + B \bullet PU/PN \qquad (8)
\end{aligned}
$$

where, according to eqns. (5) and (6):

$$
\begin{aligned}
A &= PR/PN = P_{vl0}/PN + P_{sl0}/PN \qquad (9a)\\
B &= 1/\eta^* - PR/PN \qquad (9b)\\
&= 1 + P_{fl0}/PN - P_{vl0}/PN\\
&= l/\eta_b \bullet (1 + P_{sl0}/PN) - P_{sl0}/PN - P_{vl0}/PN
\end{aligned}
$$

$$1/\eta^* = 1/\eta_b \bullet (1 + P_{sl0}/PN) \qquad (10)$$

This result is not accurate for condensing boilers, because these present a difference between η_0 and η^* which can be larger than 15%. Two kinds of condensing boilers can be considered depending on whether PR/PN is negligible or not. For condensing boilers having a very low level of stand-by losses PR/PN=0, eqn. (7) leads to:

$$\eta_c = \eta_0 \qquad (11)$$

As represented in Fig. 4, this efficiency is not a linear relationship because η_0 varies with the return water temperature.

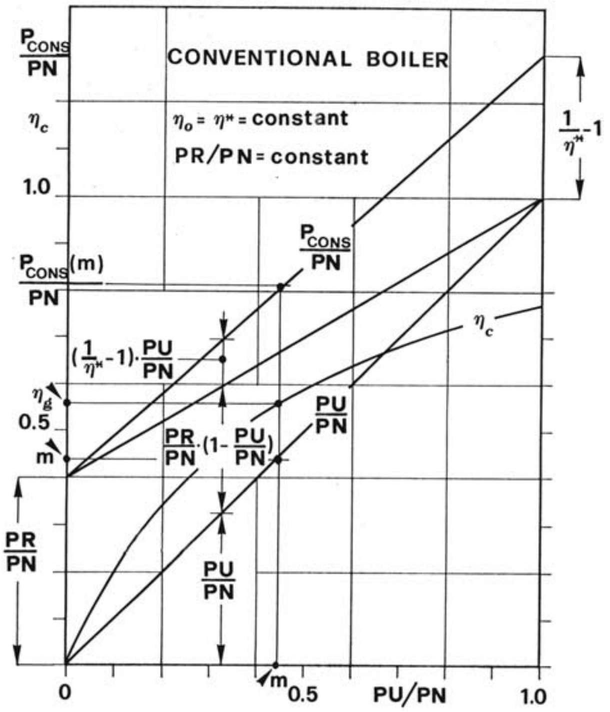


Figure 3. Acceptable model of a boiler at partial load; acceptable when temperatures are more or less constant, using stand-by losses PR and steady efficiency η_0

Not all condensing boilers have negligible stand-by losses, especially when the evacuation of flue gases takes place through the natural draught. This occurs only if the flue gas channels are very long and characterized by low pressure drops. For these boilers the relationship used was that of eqn. (7), which, obviously, is not linear and it depends on η_0, PR and the dependence law of the boiler temperatures TR and TM from load PU/PN (determined by the regulation system).

The P_{cons}/PN curve in the case of a condensing boiler connected to the plant by a 3-way mixing valve is represented in Fig. 2, where, again, the relationship is not linear

If the PR curve is unknown, the cyclical efficiency by eqn. (7) cannot be directly calculated. A straight line from the losses at zero load (point X) to the energy consumption at full load (point Y) can be assumed and the values of the constants A and B are:

$$A = PR_X/PN \qquad (12)$$
$$B = 1/\eta_{0y} - PR_X/PN \qquad (13)$$

and the linear relationship is:

$$\begin{aligned} P_{cons}/PN &= 1/\eta_c \bullet PU/PN \qquad (14)\\ &= PR_X/PN + (1/\eta_{0y} - PR_X/PN) \bullet PU/PN\\ &= PR_X/PN \bullet (1 - PU/PN + 1/\eta_{0y} \bullet PU/PN) \end{aligned}$$

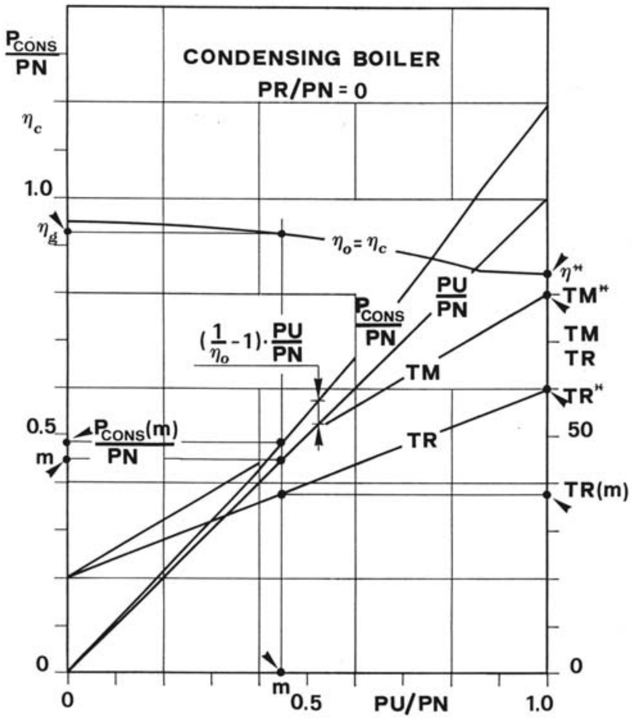


Figure 4. Model, using stand-by losses PR and steady efficiency η_0 of a condensing boiler without stand-by losses, directly connected to a plant having $TR_d = TR^*$.

The loss PR_X/PN at PU/PN=0 has to be evaluated at boiler temperature controlled by the regulation system at this load (for direct connection $PR_X/PN=0$). Values of P_{cons}/PN at the point Y can be obtained from laboratory. The efficiency η_{0y} at full load assumes the η_0 value corresponding to the return water temperature TR_y controlled by the system (the design return water TRy=TRp): $\eta_{0y}=\eta^*$ only if TRy=TRp =TR*.

5. SEASONAL EFFICIENCY

The amount of natural gas consumption, in an old apartment rented to students, was collected during an eight-year period.

This apartment measures 100 m^2 (AREA), is located in a multi-unit building and heated by an independent boiler (high efficiency boiler, 22 kW).

The students who lived in the apartment changed over the years. Their presence during the week was not constant (5 or 7 days) and their choice of the room temperature (18-22 °C) by the heating system control was different too.

The gas consumption data analysed are those going from September to June.

The total annual consumption GASy = (828 - 2250) m^3 and the profile of GAS consumption, beginning from September, are shown in Fig. 5.

The variation of GAS during the season is evident and the normalized total consumption per floor area GASy/AREA is very variable between 8.2 and 22.5 m^3/m^2.

Fig. 6 describes the adimentional values of GAScons/GASy, which is variable only during the season and can be represented by a simple sinusoidal equation (15).

Its derivative is the load LU=PU/PN of the boiler. As a result, during a season, maximum loads LUmax of the boiler are required in a short period during the winter and LUmax values can be much lower than 100 %. In the other months LU can be assumed to vary from zero to LUmax with a cosinus law (16).

$$\begin{aligned} &GAScons/GASy = \\ &= 50*(1+\sin(days - days0/2)*6{,}28/365/1{,}3))[\%] \qquad (15)\end{aligned}$$

$$\begin{aligned} &LU/LUmax= \\ &=100*\cos((days-day0/2)*6{,}28/365/1{,}3)[\%] \qquad (16)\end{aligned}$$

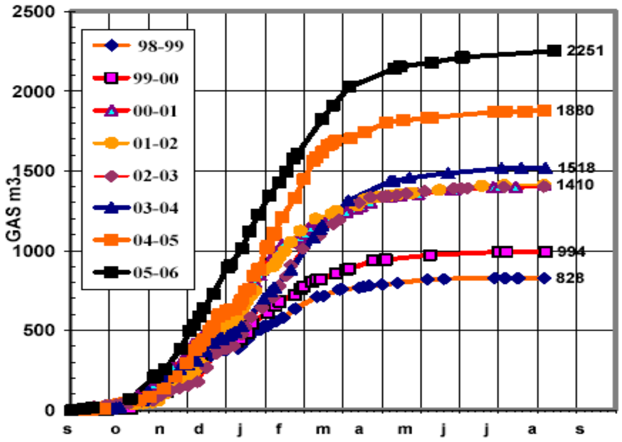


Figure 5. Profile of consumption of GAS

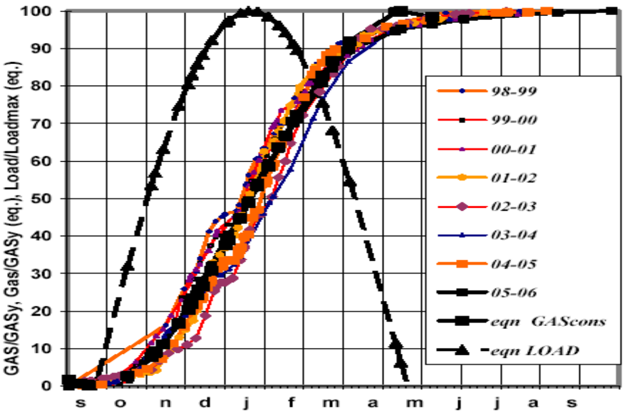


Figure 6. Profile of adimentional of GAS/GASy and LU/LUmax.

The seasonal efficiency η_g may be assumed to be equal to the cyclical efficiency η_c calculated at the mean seasonal value *m* of the loads LU = PU/PN only if the efficiency is constant:

$$\eta_g = \eta_c(m) \tag{17}$$

In order to predict the seasonal efficiency of a boiler η_g an accurate result can be gained by using the following procedure:

(1) two experimental curves representing the boiler efficiency at full load and losses must be determined by laboratory tests as function of water temperatures:
η_0(TR) and
PR((TR+TM)/2).
These two curves do not depend upon the regulation system;

(2) establish both the return TR and the delivery TM temperatures as function of the loads LU in relation to the adopted regulation system;

(3) construct the curve of the cyclic efficiency η_c as function of loads LU;

(4) use N+1 loads from 0, 1/(N)*LUmax, 2/(N)*LUmax, … to LUmax. As an example, given N=4: (0, 0.25, 0.5, 0.75, and 1)*LUmax, calculate the corresponding values of cyclic efficiencies for the boiler;

(5) consider a sinusoidal variation of loads during the season from zero to LUmax and calculate N intervals of days corresponding to the same loads: T1, T2, T3 and T4;

(6) Use the equation (18)

$$\eta g = \frac{\sum_{i=0}^{N-1} \eta_c \cdot LU \cdot T + \sum_{i=1}^{N} \eta_c \cdot LU \cdot T}{\sum_{i=0}^{N-1} LU \cdot T + \sum_{i=1}^{N} LU \cdot T}. \tag{18}$$

6. CONCLUSIONS

Sinusoidal loads from zero to maximum load (less than 100%) during a season seem to be acceptable in apartment heating plants. The relationship between energy consumption end energy demand in cyclical operation can be used with a sufficient degree of precision to calculate the seasonal consumption as a function of the sinusoidal load during the season.

A relationship of the cyclic efficiency of the boiler and of the regulation system can be calculated using two experimental curves: the efficiency at full load and the stand-by losses.

This relationship can be used for condensing boilers and for other kinds of boilers, and for the usual regulation systems as soon as the appropriate values of the constants have been experimentally defined. The evaluation of the seasonal efficiency of a boiler requires the knowledge of the efficiency at some loads.

By comparing the seasonal efficiencies it can be assumed that:

- The seasonal efficiency of a condensing boiler can be higher by 25% more than the efficiency of traditional boilers.
- The seasonal efficiencies of condensing boilers increase as the load decreases, while in the same condition the efficiencies of traditional boilers decrease.
- When the seasonal maximum load of the boiler, LUmax, is low, the seasonal efficiency of a condensing boiler presents a further increase, while that of traditional boilers decreases.
- SO_x, NO_x, dust and soot, etc., which are the constituents of the flue gas, can be dissolved in the condensed water, and the pollutants emitted to the environment can be noticeably reduced.
- It is of great significance both to environmental protection and energy saving to utilize condensing boilers.

ACKNOWLEDGMENTS

This work has been supported by Italian Ministero Istruzione Università Ricerca MIUR funds. Their contribution is gratefully acknowledged.

REFERENCES

European Commission, 1997. European methodology for the evaluation of environmental impact of buildings, *REGENER project final report, 2 Application of the life cycle analysis to buildings*, 145 p.

Eurostat, 2007. Eurostat yearbook 2006-07, *Statistical Office of the European Communities*.

Prek M., 2004. Environmental impact and life cycle assessment of heating and air conditioning systems, a simplified case study.

Rosa L., Tosato R., 1988. A simplified evaluation of the seasonal efficiency of boilers, *Proceedings of the 23rd IECEC*, Denver, CO, U.S.A..

Rosa L., Tosato R., 1985. Rendement stationnaire et rendement global d'exploitation des chaudières à gaz pour chauffage civil, *Proceedings of the CLIMA2000*, Copenhagen, Denmark,

Natural Resources Canada's Office of Energ.Efficiency (OEE), 2006, http://oee.nrcan.gc.ca/industrial/technical-info/

TOWARD AN INTEGRATION OF RISK ANALYSIS IN SUPPLY CHAIN ASSESSMENT

Jean-Claude Hennet[a], **Jean-Marc Mercantini**[b], **Isabel Demongodin**[c]

LSIS (Laboratoire des Sciences de l'Information et des Systèmes),
Domaine Universitaire de Saint-Jérôme, Avenue Escadrille Normandie-Niemen,
13397 Marseille, cedex 20, France

[a]jean-claude.hennet@lsis.org, [b]jean-marc.mercantini@lsis.org, [c]isabel.demongodin@lsis.org

ABSTRACT

This study analyzes the risks incurred by supply chains, both externally and internally. Several alternatives are presented to evaluate the risks of disruption of a supply chain and bankruptcy of one or several of its member enterprises. The issue is then to integrate risk management within classical supply chain management approaches. The difficulty of this integration is described through contradictions between Supply Chain Management and Risk Management in objectives and practice.

Keywords: vulnerability, hazard, risk analysis, risk management

1. INTRODUCTION

According to Christopher (1998), a supply chain is a "network of organizations that are involved, through upstream and downstream linkages, in the different processes and activities that produce value in the form of products and services in the hands of the ultimate customer". Some of these linkages are physical, to carry flows of products. Others are informational, to exchange messages and negotiate terms of trade.

A supply chain can also be viewed as a virtual system subject to possible dynamic reconfigurations, through arrival or departure of partner enterprises. As for any physical system, a supply chain is prone to accidents, undesirable events leading to consequences or damages on vulnerable targets. The origin of undesirable events can be related to political, social, economical, natural, technological, or organizational aspects. However, the reconfigurable nature of supply chain probably explains why risk analysis is not currently considered a leading approach in supply chain design and supply chain management (SCM). In the SCM literature, risks are often under evaluated and treated as one among many factors in economic evaluation. Typically, risk is defined as the cost of supply-demand mismatches (Ülkü, Toktay, and Yücesan 2007).

This study proposes to identify some risks incurred by supply chains, apply a risk management approach and attempt to conciliate this approach with the more traditional Supply Chain Management techniques.

2. RISKS IN SUPPLY CHAINS

The concept of risk is crucial in many sectors of todays

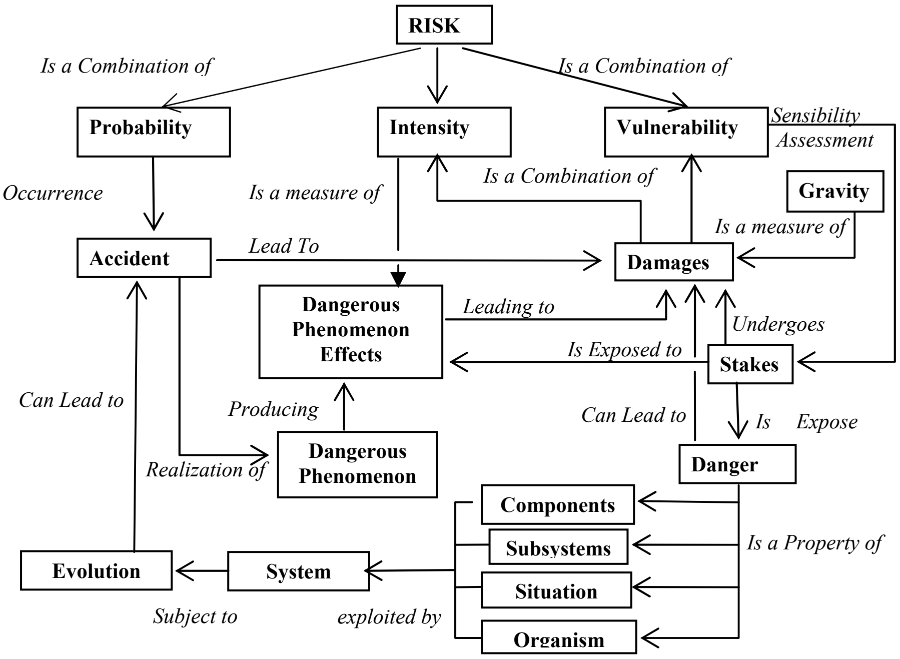


Figure 1: Some concepts and relations to define risks

Society: health, environment, economy, finance. Several international incentives have concurred to clearly define this concept.

2.1. Some definitions

According to the ISO-IEC (2002), risk can be defined as the combination of the probability of an undesired event (accident) and its consequences (damages). The notions of hazard and accident are linked with the presence of stakes that are vulnerable to the consequences of an accident. Without a stake carrying a vulnerability property, the risk related to an event is null. As a consequence, an accident is defined as the accomplishment of a dangerous phenomenon, combined with the presence of vulnerable targets exposed to the effects of this phenomenon. Some concepts and the links between them are represented in figure 1.

2.2. Vulnerability of supply chains

2.2.1. The virtual nature of supply chains

As a virtual entity non reducible to its constitutive enterprises, a supply chain seems at first sight to keep away from the structural and operational vulnerability of traditional corporate firms. Risks seem to be located in the enterprises rather than in the network that they constitute.

In its design stage the structure and components of a supply chain can be easily modified. Hence, a simple risk reduction measure consists in replacing a risky firm by a less risky one. However, risk then becomes a key factor in choosing the partners of a supply chain. And conversely, firms may become reluctant to participate to a supply chain if they correctly assess the risk of being pushed away from partnership precisely when they encounter difficulties and would rather need support.

2.2.2. SMEs in supply chains

A major problem in the design and organization of supply chains is heterogeneity of the enterprises involved. In terms of risks, Finch (2004), has clearly shown that SMEs (Small and Medium Enterprises) greatly increase their exposure to risk when becoming part of a supply chain. It is clearly more risky for an SME to invest all its assets and use investment loans than for a large company to devote a marginal part of its resources for a joint venture.

As noted in (Hennet 2008), SMEs are often considered as the weakest links in supply chains and many of them have started to realize the risk of their involvement in a global supply chain. From the very definition of a supply chain, its main stake and reason for being is to provide goods and/or services in a cheap, efficient and profitable manner.

Failure to fulfill any of these requirements on a permanent basis inevitably leads to disruption of the supply chain. And if this is not a relevant risk for a virtual entity, this is a real risk for companies that have invested a significant part of their resources in setting up the supply chain.

2.2.3. Complexity as a source of vulnerability

Increasing complexity of supply networks that connect suppliers, manufacturers, distributors, retailers, and customers, leads to more interdependence between firms. At each stage of a supply chain, different types of social, environmental and economical risks will affect system performance. Ultimately, user satisfaction will determine the viability of the supply chain.

Indeed the probability that something happens at a particular node or connection is higher than for a small and simple network.

2.2.4. Disturbance amplification

The literature on supply chains, and specially studies devoted to existing supply chains, provides a valuable source in risk identification. As an example, the literature on the bullwhip effect has clearly identified as a dangerous phenomenon the amplified fluctuation of inventory levels upward a supply chain (Lee, Padmanabhan, and Whang 1997; Towill 2005).

The second main risk that can be generated by uncertainties within the supply chain has been called the “chaos” risk (Childerhouse et al. 2003; Li and Hong 2007). This risk is mainly characterized by a high probability of strong distortions on key information and critical decisions

Another example of a phenomenon that has proven itself dangerous for a supply chain is the fact that market has put a lot of pressure on firms to differentiate their products. This has led companies to rely on several third parties and has consequently increased the risks by increasing interactions and complexity in the supply chain.

2.3. Risk analysis

Risk analysis firstly requires a methodology for identifying all the risks applying to the considered system, a supply chain, along its life-cycle. Then, each risk is associated with a system vulnerability that can be revealed through scenarios to be simulated to measure risk intensity.

Risks that can lead to supply-chain disruptions are as different as natural catastrophes, strikes, political instability, fires or terrorism. Vulnerability of supply chains to these risks has increased because of modern practices such as lean management and just-in-time inventory.

Taxonomy of supply chains risks can be derived from a careful scanning of internal and external hazards. A classification can be obtained from the different views, roles and activities of the system in its environment. As a starting example, supply chain disruptions can arise:

- from external sources - such as natural phenomena, financial disturbances (changes in exchange rates, taxes), social movements (labor strikes, new regulations), economical problems (unavailability of some product).
- from internal sources - such as products and processes in use at the different stages, design

and dimensioning of the supply chain, failure to integrate all functions in a supply chain, synchronization of product flows, qualitative or quantitative policies at the different stages.

The Risk Breakdown Structure (RBS) technique can be used for supply chains as it has shown its efficiency in many sectors of activity, such as software development projects (Kwak and Stoddard 2004), construction design (Chapman 2001).

MIT research group on “Supply Chain Response to Global Terrorism” has shown that firms usually focus on the type of disruption and not its source in order to know how to prepare against risks. What is important is the type of “failure modes, i.e. the limited ways in which the disruption affects the supply-chain”.

A disruption in supply for example can be caused by a strike, an earthquake or a terrorism action and in each case will have the same impact. The team distinguishes 6 different types of failure modes (see Table 1) that are: “Disruption in supply, Disruption in transportation, Disruption at facilities, Freight breaches, Disruption in communications, and Disruption in demand.”

Table 1: Supply-Chain Failure Modes

Failure Mode	**Description**
Disruption in supply	Delay or unavailability of materials from suppliers, leading to a shortage of inputs that could paralyze the activity of the company.
Disruption in transportation	Delay or unavailability of the transportation infrastructure, leading to the impossibility to move goods, either inbound and outbound.
Disruption at facilities	Delay or unavailability of plants, warehouses and office buildings, hampering the ability to continue operations.
Freight breaches	Violation of the integrity of cargoes and products, leading to the loss or adulteration of goods (can be due either to theft or tampering with criminal purpose, e.g. smuggling weapons inside containers).
Disruption in communications	Delay or unavailability of the information and communication infrastructure, either within or outside the company, leading to the inability to coordinate operations and execute transactions.
Disruption in demand	Delay or disruption downstream can lead to the loss of demand, temporarily or permanently, thus affecting all the companies upstream.

3. RISK MANAGEMENT

3.1. The RM process

Classically, the Risk Management (RM) process consists of a series of measures and steps to gradually decrease the risks through a decrease of their probability of occurrence and a decrease of the system vulnerability by developing resilience.

The first step after the risk analysis is to identify and correctly assess the largest set of actions that can be taken to reduce the risks. Then, the second step is to select the most appropriate risk reduction actions.

Risk Management (RM) methodology can be applied through a classical decomposition into the following stages:

- assessment of risk reduction measures,
- selection of risk reduction measures,
- risk supervision in supply chains.

The strategic issue is to determine which actions to undertake in order to manage disruption risks.

3.2. Multi-sourcing

A classical risk mitigation technique consists in introducing redundancy. As a typical application of Risk Management to supply chains, the approach developed in (Pochard 2003) relies on dual sourcing and shows that the real options concept is an adapted tool to evaluate such a strategy. It develops an analytic model to analyze and value the benefits of relying on dual sourcing. This model takes into account various parameters such as the frequency of disruption and the loss of market share.

Retailers often face random variations both on the demand and the supply side. In such cases, Arda and Hennet (2006) show that it is generally more profitable for the retailer to procure from several suppliers rather than to use a single one. Similar results apply when the retailer is concerned with supplier default risk (Babich, Ritchken, and Burnetas 2007). An increase of wholesale prices for smaller delivered quantities could balance this trend. Except that wholesale price increase is often very limited, in particular if the retailer is in a dominant position or if there is a strong competition between suppliers. A possible response in this situation is to develop a symmetric strategy, as described in (Sucky 2007) where a supplier dynamically selects his (her) retailer. However, this would be an attempt to become a dominant partner, which is particularly difficult for a small enterprise facing larger ones.

3.3. Supervision and monitoring

In terms of implementation, a frequent updating of business plans and aggregate planning are essential to organize production with reactivity and flexibility. Risk supervision should play an essential part in the monitoring policy of each enterprise that belongs to a supply chain.

In particular, if dual sourcing is part of the current strategy, managers need to monitor the usefulness of such a solution over time. Their environment may change and firms need to adapt their sourcing strategies over time.

4. CONVERGENCE BETWEEN RM AND SCM

RM and SCM methodologies are so different that a strong effort is needed to conciliate the two viewpoints. The objective of the section is to propose some guidelines toward the design and operation of efficient and sustainable supply chains.

4.1. Integration of risks in Business Process Management Techniques

4.1.1. The SCOR model

One of the leading methods for Supply Chain Management is based on the SCOR (Supply Chain Operations Reference) Model. This model provides a framework for assessing and evaluating a Supply Chain in terms of process models. Three levels of process models are distinguished (Supply Chain Council 2008). The top level contains 5 core management processes called: Plan, Source, Make, Deliver, Return. The second level is the configuration level with 3 processes: Planning, Execution, Enable. The third level details level 2 processes, often in the form of workflows.

This approach can be very useful to identify the weak links in supply chains, both (but not simultaneously) from the viewpoints of risks and effectiveness.

Integration of Risk Management in business process management techniques has recently become a major concern in Supply Chain Management. Clearly, disruptions may damage strongly the supply process and firms may loose business. Protecting the supply-chain against such events may also become a strategic advantage toward the competitors. For instance, in case several firms suffer from the same disruption, companies that are well prepared will limit their loss, recover faster and may even take market shares from their competitors.

As stated in the SCOR booklet (Supply Chain Council 2008), “the new release of the SCOR model enables a company to more effectively balance risk impact and costs of risk mitigation with overall supply-chain management costs”. Supply chain operations are described as a series of activities. Then, a Cross-Functional Process Map (CFMP) is constructed to identify and redesign non value-added activities (SCM view), and to identify and eliminate highly risky steps (RM view) (Li and Hong 2007).

4.1.2. Quality Management

An effort to conciliate supply chain effectiveness with security has been described in (Lee and Wolfe 2003). For them, quality management is the key to conciliate RM and SCM through reducing defects without increasing costs. Thus, firms must “promote measures that also increase supply-chain flexibility”. Applying the principles of this theory to supply-chain security, the authors argue that firms need to focus on prevention rather than inspection and have an advanced process control.

4.2. Toward a model-based approach to RM and SCM

4.2.1. Semi-formal models of supply chains

SCM methods may rely on models of two different types: semi-formal models such as the process models of the SCOR (Supply Chain Operations Reference) approach and analytical models, based on mathematical expressions.

Business process models have been widely used for design, evaluation and management of supply chains from the viewpoints of architecture, organization and communication. Multi-agent models also belong to the class of semi-formal models, mainly used to organize information flows in Supply Chains and improve their performance (Labarthe et al. 2006).

In risk analysis and management for supply chains, models are mainly used to simulate hazardous scenarios and evaluate the vulnerability of the whole system and its components, the partner enterprises.

Quantitative models of supply chains are better suited to describe dynamical and balance equations for product flows and cash flows.

According to Pundoor and Herrmann (2006), three types of simulation approaches have been developed: Discrete event models, specialized softwares and distributed simulation. Integration of risks in Supply Chain models requires a representation of uncertainties and unexpected phenomena. This requirement seems to indicate that stochastic discrete event models may be suitable to represent and evaluate risks in supply chains. However, such models are rather complex, especially to study transient dynamical effects of disturbances.

4.2.2. Discrete event models

In this paper, queuing networks and time-series models are proposed as candidate simulation models to represent the occurrence of hazardous phenomena and their impact on a supply chains.

As for many Discrete Event models, queuing networks models of supply chain hardly differentiate the system and its control. This is particularly true in inventory management, where dynamics of state evolution are intimately dependent upon the inventory policy. A comparative evaluation of several inventory systems would then require several different simulation models. It is thus important to combine analytical studies with simulation to select the most efficient controls in terms of profit and/or risks. In distributed inventory control, base-stock policies with respect to inventory position levels have been shown optimal and robust for typical supply chains.

According to the (S_i - 1, S_i) base-stock policy, the inventory initially contains S_i units and a unitary replenishment order is placed whenever the inventory position declines to the reorder point S_i - 1, i.e. whenever a demand occurs.

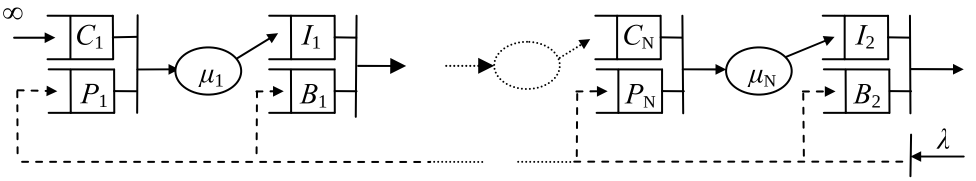


Figure 2. N-stage make-to-stock queuing system

The (S_i - 1, S_i) base-stock policy maintains the inventory position (*stock on hand – backorders + outstanding orders*) constant at the base-stock level S_i and the production facility operates if the inventory (*stock on hand - backorders*) is short of the base-stock level S_i. Whenever a customer order arrives, the manufacturer places not only one end-product processing order, but also one unitary component order to the supplier. Likewise, a demand arrival at the supplier immediately triggers a component processing order and a unitary demand of raw material.

Figure 2 illustrates the queuing network of a serial system with N manufacturing stages. Solid arrows represent material flow and dashed arrows represent information flows. For $i \in [1, N]$, P_i denotes the number of uncompleted orders waiting in the processing queue; K_i, the number of uncompleted orders in the system (P_i plus the order eventually in service); C_i, the number of units delivered from stage i-1 waiting in the processing queue of stage i; N_i, the number of units in the system (C_i plus the unit eventually in service); I_i, the number of units in the finished-goods inventory; B_i, the number of outstanding backorders.

In (Arda and Hennet 2008), the model of figure 2 has been analyzed analytically for $N = 2$, to compute the limiting performance of the system in steady state.

Disturbances can be introduced in this model through sudden changes in demand, processing and/or transportation rates λ, μ_i, or in inventory levels I_i. The influence of such disturbances is clearly difficult to analyze mathematically, but the model can be used in simulation to identify the cascade of risks generated by such changes, represented as scenarios with attached probability.

Another promising approach to model a supply chain subject to uncertainties, randomness and risks is the time series ARIMA (Auto-Regressive Integrated Moving Average) model. It is an input-output model in which the current output depends linearly on past inputs, past outputs and on current random inputs. The basic model of Box and Jenkins (1976), is monovariable. However many multivariable extensions have been constructed to forecast and control coupled systems.

The ARIMA model of Gilbert (2005) combines

- an ARMA(p, q) model of demand: Z_t with stationary mean value, μ and a random input sequence $\{a_t\}$:

$$Z_t = \mu + \sum_{i=1}^{p} \phi_i (Z_{t-i} - \mu) + \sum_{j=0}^{q} \theta_j a_{t-j}$$

- an inventory balance equation to model the supply chain:

$$I_t = I_{t-1} + O_{t-L} - Z_t$$

- with I_t the inventory at the end of period t and O_{t-L} the order placed at the end of period t-L and expected in period t.

Using this model, the author shows that the sequence of inventories $\{I_t\}$ and the sequence of orders, $\{O_t\}$ can be represented by ARIMA models. Furthermore, the parameters of these models allow quantifying the bullwhip effect along the chain.

However, until recently, such models have been used without integrating the risks aspects. A convenient manner to represent risky situations could be through identification of hazardous states. A supervisory control objective for risk avoidance would then be to consider such states as forbidden. Another technique for avoiding risks could be, as in (Hennet 2003), to maintain the system state in an invariant region contained in the domain of safe conditions.

Hazardous phenomena often occur suddenly and their intensity, which may be very high, is a key property that determines the intensity of their effect. Using the ARMA model of demand described above, the random input sequence $\{a_t\}$ can be used to represent the occurrence of one or several hazardous events.

Simulation can then be used to determine if the system can be maintained in its viable region. Scenarios are translated into pseudo-random input sequences and the system evolves under supervisory actions. A key issue is then the controllability problem in the presence of constraints on the system state.

In general, it is very unlikely that a single enterprise could resist high intensity damages without the help of its partner companies in the supply chain. In this respect, contracts can be seen as an essential tool to conciliate risk sharing with economic efficiency.

4.2.3. Contracts between supply chain partners

Contracts may reduce the risks of financial losses and even bankruptcy of Supply Chain partners. A part of the risk is then supported by the other partners and by the supply chain itself. Multi-sourcing and multi-selling can reduce the risk that an accident occurring in the supply chain would propagate to all the other partners through non satisfaction of final customers. However, for such diversifications not to generate instability in demand, it is necessary to establish trade contracts such as the ones proposed in Cachon and Larivière (2001) between a manufacturer and his suppliers. Under such contracts, orders are divided in 2 parts: firm commitments and options. Firm commitments clearly guarantee a minimum level of revenue to the suppliers, while satisfaction of the manufacturer's maximum order level forces the supplier to install a sufficient capacity.

5. CONCLUSIONS

A major improvement in the understanding of Supply Chains has recently emerged from setting into evidence the importance of risk analysis and management for the survival of a supply chain and its member enterprises. Beyond the recent integration of risk analysis in the leading approach for Supply Chain Management, namely the SCOR model, new models and new tools are needed to help firm managers. Key issues are internal and external hazard identification, vulnerability assessment, construction of risk mitigation policies, and a persistent effort to supervise the system to maintain it as far as possible from its most risky boundaries.

REFERENCES

Arda, Y., Hennet, J.C., 2006. Inventory control in a multi-supplier system. *International Journal of Production Economics,* 104(2), 249–259.

Arda, Y., Hennet, J.C., 2008. Inventory control in a decentralized two-stage make-to-stock queueing system. *International Journal of Systems Science,* 39(7), 741–750.

Babich, V., Ritchken, P. H., Burnetas, A. N., 2007. Competition and Diversification Effects in Supply Chains with Supplier Default Risk. *Manufacturing & Service Operations Management*, 9(2), 123–146.

Box, G.E.P, Jenkins, G.M., 1976. *Time series analysis: forecasting and control*. San Francisco: Holden-Day.

Cachon, G., Lariviere, M., 2001. Contracting to Assure Supply : How to Share Demand Forecasts in a Supply Chain. *Management Science,* 47(5), 629-646.

Chapman, R.J., 2001. The controlling influences on effective risk identification and assessment for construction design management. *International Journal of Project Management,* 19 (3), 147–160.

Childerhouse, P., Hermiz, R., Mason-Jones, R., Popp, A., Towill, D., 2003. Information flow in automotive supply chains. *Present Industrial Practice,* 103(3), 137–149.

Christopher, M., 1998. *Logistics and Supply Chain Management – strategies for reducing costs and improving service*. 2nd ed. London: Financial Times: Pitman Publishing.

Finch, P., 2004. Supply chain risk management. *Supply Chain Management*, 9(2), 183–196.

Gilbert, K., 2005. An ARIMA supply chain model. *Management Science,* 51(2), 305–310.

Hennet, J.C., 2003. A bimodal scheme for multi-stage production and inventory control. *Automatica,* 39, 793–805.

Hennet, J.C., 2008. SMEs in supply chain networks: cooperation and competition. In: Villa, A., Antonelli, D., eds. *A Road Map to the Development of European SME Networks: Towards a Collaborative Innovation.* Elsevier, (to appear).

ISO-IEC guide 73, 2002. http://www.iso.org.

Kleindorfer, P.R., Saad G.H, 2005. Managing Disruption Risks in Supply Chains. *Production and Operations Management*, 14(1), 53–68.

Kwak Y.H., Stoddard J, 2004. Project risk management: lessons learned from software development environment. *Technovation,* 24, 915–920.

Labarthe, O., Ferrarini, A., Espinasse, B., Montreuil, B., 2006. Multi-agent modelling for simulation of customer-centric Supply Chain. *International Journal of Simulation and Process Modelling,* 2(3/4), 150–163.

Lee, H.L., Padmanabhan, V., Whang, S., 1997. Information distortion in a supply chain: the bullwhip effect. *Management Science*, 43(4), 546–558.

Lee, H. L., Wolf, M., 2003. Supply chain security without tears. *Supply Chain Management Review,* 7 (1), 12–20.

Li, J., Hong, S-J., 2007. Towards a New Model of Supply Chain Risk Management: the Cross-Functional Process Mapping Approach. *International Journal of Electronic Customer Relationship Management*, 1(1), 91–107.

Pochard, S., 2003. *Managing Risks of Supply-Chain Disruptions: Dual Sourcing as a Real Option*. Thesis (Master), MIT.

Pundoor, G., Herrmann, J.W., 2006. A hierarchical approach to supply chain simulation modelling using the Supply Chain Operations Reference model. *International Journal of Simulation and Process Modelling,* 2(3/4), 124–132.

Sucky E., 2007. A model for dynamic strategic vendor selection. *Computers and Operations Research*, 34, 3638–3651.

Supply Chain Council, 2008. SCOR booklet, version 9.0.

Towill, D.R., 2005. The impact of business policy on bullwhip induced risk in supply chain management. *Int. Journal of Physical Distribution and Logistics Management*, 35(8), 555–575.

Ülkü, S., Toktay, L.B., Yücesan E., 2007. Risk Ownership in Contract Manufacturing. *Manufacturing & Service Operations Management*, 9(3), 225–241.

SUPPLY CHAIN MANAGEMENT THROUGH P AND PI CONTROLLERS

Carlos Andrés García, Pedro Balaguer, Ramón Vilanova

Autonomous University Of Barcelona
Department of Telecommunication and Systems Engineering
Bellaterra, 08193, Spain
CarlosAndres.Garcia@uab.cat

ABSTRACT

An important phenomenon in supply chain management, know as the bullwhip effect, suggests that demand variability increases as one moves in the supply chain. In this paper, a discrete time model for the beer game is derived by using the well-known z-transform. The system can be viewed as linear discrete MIMO system with lead times and integrators. We analyze the impact of P and PI controllers as replenishment policies. We also analyze the stability of the system using the characteristic equation and Jury criterion. Finally, a comparison between our proposed control-based strategy and other existing replenishment policies is performed.

Keywords: supply chain, beer game, z-transform
bullwhip effect, PI control.

1. INTRODUCTION

Supply chain management has attracted much attention among process system engineering researchers recently. A supply chain includes all the participants and processes involved in the satisfaction of customer demand: transportation, storages, wholesales, distributors and factories Dejonckheere, Disney, Lambrecht and Towill (2002). A large number of participants, a variety of relations and processes, dynamics and the randomness in material and information flow prove that supply chains are complex systems in which coordination is one of the key elements of management. In this paper the focus is on the analysis and control of the material balance and information flow among of the system.

An important phenomenon in supply chain management, known as the bullwhip effect, suggests that demand variability increases as one goes up in the supply chain Hoberg, James, Bradley, Ulrich and Thonemann. (2007). The causes of the bullwhip effect can be due to the forecasting demand, the lead times, order batching, supply shortages and price fluctuations. We will mainly address the non-zero lead times and particularly the forecasting demand.

In order to demonstrate the existence of this effect the beer game was created at the beginning of the sixties School of Management, Massachusetts Institute of Technology (MIT) Dragana, Panić and Vujošević (2007). The game simulates a multi-echelon serial supply chain consisting of a Retailer (R), a Wholesaler (W), a Distributor (D) and a Factory/Manufacturer (M).

We model the basic protocol of the "beer distribution game". The mathematical model used is the transfer function which represents the relation between the input and output of a linear time invariant system (LTI). For a continuous time domain models it is customary to conduct a theoretical analysis using the Laplace transform to convert ordinary differential equations into s-domain transfer functions. In this case we are dealing with discrete signals and systems, therefore our theoretical analysis is achieved with the z-transform.

After the system is modelled one obtains a transfer function of one chain echelon introducing an ordering strategy based on controller design principles. In this work P and PI control structures are proposed and tuned accordingly, to eliminate the bullwhip effect.

We analyse the P and PI controllers parameter range for which the system is stable based on the characteristic equation and the Jury criterion. We also examine the effect of demand forecasting and lead time in such a system using the z-transform technique.

Finally in section 4 we observe the mathematical structure of replenishment strategies approached by other authors Marko and Rusjan (2008), in order to classify them according to the laws of implicit control and the type of feedback that is employed.

2. BEER GAME MODEL

Lets us consider a basic beer supply chain as shown in figure 1. There are four logistic echelons: Retailer (R), wholesales (W), distributor (D), and Factory (F).

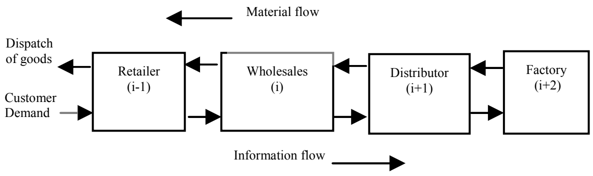


Figure 1. The Block Diagram Of Supply Chain.

Let $I_i(k)$ denote the inventory level of a echelon of chain *I(R,W,D,F)* at any discrete time instant *k*. We also let $Y_{i,i-1}(k)$ indicate the amount of goods to be delivered to node *i*-1 by the upstream node *i* at the instant *k*, $O_{i-1,i}$ indicates the demand received by node *i* from downstream *i*-1. A time delay of L due to transport is assumed for all delivery of goods so that goods dispatched for an upstream $(i+1)$ at time *k* will arrive at time *k*+l at node *i*. However, due to the need for examination and administrative processing, this new delivery is only available to the node *i* at k+L+T_0. The orders placed in the upstream for the *i* node is denoted for $O_{i,i+1}(k)$. We assume that the upstream $i+1$ supplier has sufficient inventory so that the orders of node *i* are always satisfied so that the amount of goods delivered by upstream $i+1$ in instant *k* $Y_{i+1,i}(k)$ is equal to the order made for the downstream *i* in a previous time $O_{i,i+1}(k-L-T_O)$. It's also assumed the amount of goods delivered by upstream *i* in *k* instant $Y_{i,i-1}(k)$ is equal to the order made for the downstream *i*-1 in a previous time $O_{i-1,i}(k-L-T_O)$.

The result of the integration of the difference between amount goods that entry from upstream node $i+1$ and amount goods that dispatched for downstream node *i*-1 is known as inventory balance, this has a role as a buffer to absorb the demand variability. In other words, the inventories should have stabilizing effect in material flow patterns. The equation for inventory balance at node *i* is given by:

$$I_i(k) = I(k-1) + O_{i,i+1}(k-L-T_0) - O_{i-1,i}(k) \qquad (1)$$

A signal control denoted by $U_{i,i+1}$ is the result of a control strategy for compute the orders to upstream $O_{i,i+1}$. For example, a simple P-control can be used hence $U_{i,i+1} = K_p\left(I_0(z) - I_i(z)\right)$. We assume that ordering information is communicated after a time delay T_o. Hence, the order placed by the node *i* at the upstream $i+1$ is given by:

$$O_{i,i+1}(k) = U_i(k-T_o) \qquad (2)$$

The z-transform is a powerful operational method when one works with discrete control systems because the differential equation is converted to an algebraic problem. Using the Time Shifting z property $Z(x(k-n)) = z^{-n}X(z)$ Ogata (1996), on equations (1) and (2), we obtained the z-transform of the above discrete time model, this is given by the equations (3) and (4).

$$I_i(z) - I_i(z)z^{-1} = U_{i,i+1}(z)z^{-(T_o+L)} - O_{i-1,i}(z) \qquad (3)$$

$$O_{i,i+1}(z) = U_{i,i+1}(z)z^{-T_o} \qquad (4)$$

It is considerer as a MIMO system and its matrix transfer function shown in equation 6.

$$\begin{vmatrix} O_{i,i+1}(z) \\ I_i(z) \end{vmatrix} \quad \begin{vmatrix} 0 & z^{-To} \\ -\dfrac{z}{z-1} & \dfrac{z}{z-1}\left(z^{-To}z^{-L_i}\right) \end{vmatrix} \quad \begin{vmatrix} O_{i-1,i}(z) \\ U_{i,i+1}(z) \end{vmatrix} \qquad (5)$$

The corresponding simplified block diagram is given in figure 2.

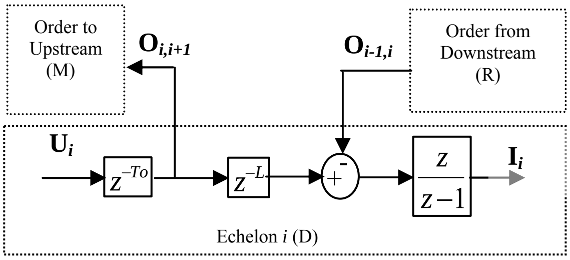


Figure 2. The Block Diagram Of Node *I* Of A Supply Chain.

3. ANALYSIS OF THE EFFECT OF P AND PI CONTROLLERS AS REPLENISHMENT INVENTORY

A control system is a combination of elements (components of the system) which enable us to control the dynamics of the selected process in a certain way. The PID (proportional, integral and derivative) is the controller most commonly used in control engineering because of its flexibility and simplicity. Therefore there will by introductory analysis of this controller as inventory replenishment policy. In this section we analyze the effect of each action (proportional, integral) over the stability in close loop in one level of the supply chain. We perform simulations with different values of parameters (K_p, K_i) and finally we compare the behaviour of the inventory and orders signals using P or PI actions.

3.1 Proportional Control

In this section we obtain the transfer function of the system in close loop with the proportional action, we approach a proportional controller and we analyse the stability of the system and the behaviour of the orders $O_{i,i+1}$ an the inventory level I_i. Our interest is management the of inventory level therefore we uses a feedback-level inventory and a proportional controller. The corresponding simplified block diagram is show below.

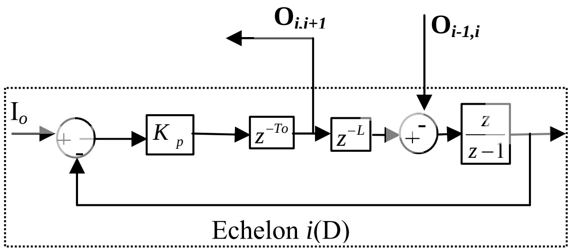


Figure 3. The Block Diagram Of Node *I* Within The Supply Chain With P-Controller.

The orders to upstream $O_{i,i+1}$ is given by the following expression.

$$O_{i,i+1}(z) = K_p z^{-To}(I_0(z) - I_i(z)) \quad (6)$$

Where K_p the proportional parameter that multiplicity the difference between the desired inventory $I_0(z)$ and physical inventory $I_i(z)$.

The physical inventory is given by:

$$I_i(z) = O_{i,i+1}(z) z^{-L}\left(\frac{z}{z-1}\right) \quad (7)$$

Replacing $I_i(z)$ in (7) obtains the closed loop transfer function or the rate ordering can be derived as the following equation.

$$G_o(z) = \frac{O_{i,i+1}(z)}{I_0(z)} = \frac{K_p z^{L-1}(z-1)}{z^{L+T_o} - z^{(L+T_o)-1} + K_p}$$
(8)

Similarly we obtain the transfers function that relates the physical inventory with the inventory desired.

$$G_I(z) = \frac{I_i(z)}{I_0(z)} = \frac{K_p(z-1)}{z^{L+T_o} - z^{(L+T_o)-1} + K_p} \quad (9)$$

Both relations contain the same characteristic equation, hence it is possible to do stability analysis for both transfer functions using this equation.
The characteristic equation is given below:

$$z^{L+T_o} - z^{(L+T_o-1)} + K_p = 0 \quad (10)$$

3.2 Stability Analysis

The objective of this section is to examine some cases on supply chain operations with a proportional control of inventory levels. A system is stable if all the roots of the characteristic equation lie within the unit circle.

$|z_i| < 1$ Stable.

$|z_i| = 1$ Marginally stable. (11)

$|z_i| > 1$ Unstable

The Jury stability criterion Ogata (1996), is essentially the discrete time analogue of the continuous time Routh stability criterion. It is a technique for verifying the stability of a linear discrete time system described in the z-domain. This is used directly to characteristic equation without evaluating the roots.

The Jury criterion is to check that four conditions are met.

Condition 1: If F(z) is a function of integer n grade $F(z) = a_0 z^n + a_1 z^1 ......a_n z^n$, which represents the characteristic equation of the transfer function, which must be fulfilled:

$$F(1) > 0 \quad (12)$$

Applying this condition in the characteristic equation (10):

$$(1)^{T_o+L} - (1)^{T_o+L-1} + K_p > 0 \quad (13)$$

As a result:

$$K_p > 0 \quad (14)$$

Equation (14) tells us that the values of K_p must by positive.

Condition 2: When the value (L+ To) is even number F(-1)>0, the result of applying this approach will be:

$$(-1)^{To+L} - (-1)^{To+L-1} + K_p > 0 \quad (15)$$

It follows that:

$$K_p > -2 \quad (16)$$

Condition 3: When the value (L+ To) be odd the function F(-1)<0, the result of applying this approach will be:

$$(-1)^{T_o+L} - (-1)^{T_o+L-1} + K_p < 0 \quad (19)$$

That is same as:

$$K_p < 2 \quad (20)$$

Condition 4: this result is immediate

$$|a_n| < a_0 \quad (21)$$

Hence the result is show in equation (22).

$$K_p < 1 \quad (22)$$

This value is more restrictive than the obtained in equation (20).

Using the conditions 1 and 4 we can find the range of values of K_p in which the system is stable. This range is given by:

$$0 < K_p < 1 \quad (23)$$

In this case for our simulations we chose a set point of 10000 units, with values $T_0=1$ sample and L=2 samples. Figure 4 shows the orders placed in the upstream for the *i* node $O_{i,i+1}(t)$ and the inventory level of the echelon of chain $I_i(k)$ for different values of K_p. We can see that the orders and inventory signals are stable for $K_p < 1$ and unstable for $K_p \geq 1$. With the proportional controller is possible to stabilize the system whit certain values of K_p but a phenomenon (offset) that is a mistake of steady state is inevitable because if the error is constant the control action is constant too.

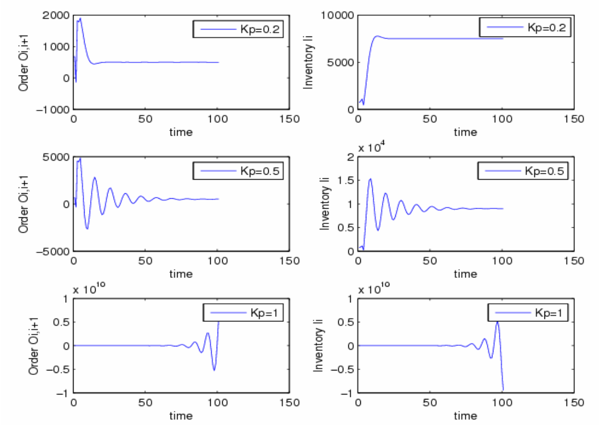


Figure 4. Simulation Results Of A Supply Chain Unit With A P Controller Using Different Values Of K_p And Stochastic Demand From Downstream.

3.4 Transfer Function With PI Actions

The integral action has some characteristics that improve the response of the system in close loop. One of these properties is that it removes the offset because the control increases although the error remains constant (integrates the error), hence we analyze the behaviour in a supply chain. In this section we show the transfer functions of the system in close loop with the proportional and integral action introducing the PI controller.

The discrete transfer function of PI controller with sample period T=1 sample is given by:

$$C(z) = K_p + \frac{K_i}{z-1} \tag{24}$$

Where K_p is the proportional constant and K_i is the integral constant.

The transfer function in closed loop of the system which relates to orders delivered by upstream is:

$$\frac{O_{i,i+1}(z)}{I_0} = \frac{K_p z^L + (K_i - K_p) z^{L-1}}{z^{L+T_o+1} - 2z^{L+T_o} + z^{L+T_o-1} + K_p z - K_p + K_i} \tag{25}$$

Similarly, the transfer function in closed loop of the system which relates the inventory balance and the desired inventory is:

$$\frac{I_i(z)}{I_0} = \frac{K_p z - K_p + K_i}{z^{L+T_o+1} - 2z^{L+T_o} + z^{L+T_o-1} + K_p z - K_p + K_i} \tag{26}$$

The orders of both functions depend of lead time and we can see it is possible to tune the PI parameters in order to stabilize the system.

The characteristic equation is given by:

$$z^{L+T_o+1} - 2z^{L+T_o} + z^{L+T_o-1} + K_p z - K_p + K_i = 0 \tag{27}$$

The basic block diagram is shown in figure 5.

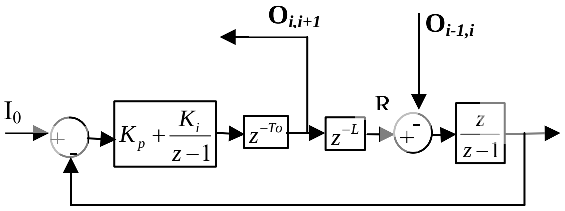


Figure 5. The Block Diagram Of Node *I* Of A Supply Chain With Pi-Controller.

The goal of this section is to examine some asymptotic cases of the supply chain operations with a proportional integral control PI of the inventory levels. Although the lead time is a parameter that can vary depending on the characteristic of each chain, we can through Jury criterion estimate a range of K_p and K_i around which the system is stable. $0 < K_p < 1, K_i < 0.35K_p$.This asymptotic analysis provides useful insights.

Using matlab we simulate different values of K_i in order to observer the influence over the stability of the system. Using the order policy defined by Eqs. (25) And (26) with values $K_p=0.2$, $T_0=1$, and L=2.

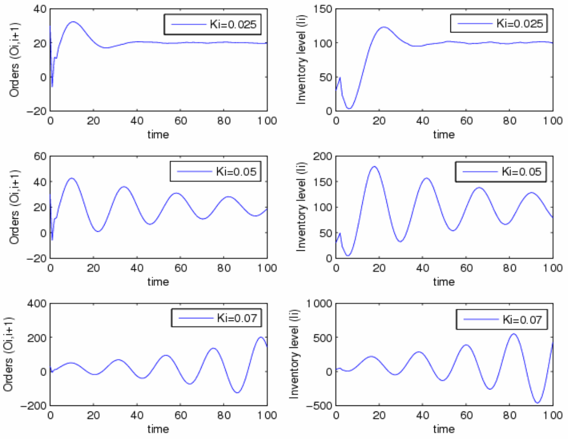


Figure 6. Simulation Results Of A Supply Chain Unit With A PI Controller Using Different Values Of Ki And Stochastic Demand From Downstream.

This allows us to conclude that it is possible through tuning PI to stabilize the pattern of the orders and thus avoid the bullwhip effect. It eliminates offset by which allows us to control inventory, which translates into a major supply chain management.

4. BULLWHIP REDUCTION

We are interested in the ratio of amplitude of the generate orders ($O_{i,i+1}$) over the amplitude of demand ($O_{i-1,i}$) that is know as ratio amplitude (AR) and this is a measurement of bullwhip Dejonckheere, Disney, Lambrecht and Towill (2002). Some new metrics for the bullwhip effect are introduced specifically based on the Frequency Response (FR). The Fourier transform, is an algebraic method of decomposing any time series into a set of pure sine waves of different frequencies, with a particular amplitude and phase angle associated with each frequency therefore through of FFT over (AR) we can measure the bullwhip effect. In this case we use a step signal plus white noise (σ =0.1) added to represent the demand. The FR is given in figure 7 for the order policy defined by Eqs. (10) And (11) with values T_0=1 sample, and L=2 samples.

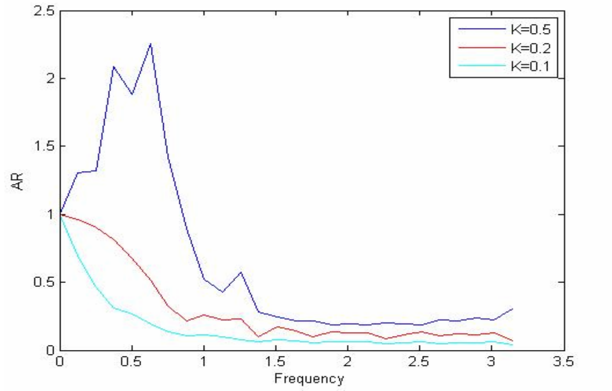


Figure 7. FR For The P Controller As Replenishment Rule.

We can see that exist a reduction of bullwhip using K_p<0.5.

The figure below shows the amplitude ratio (RA) where it is observed that for values of K_p = 0.2 and K_i = 0.02 presents a demand smoothing and thus reduction of Bullwhip effect.

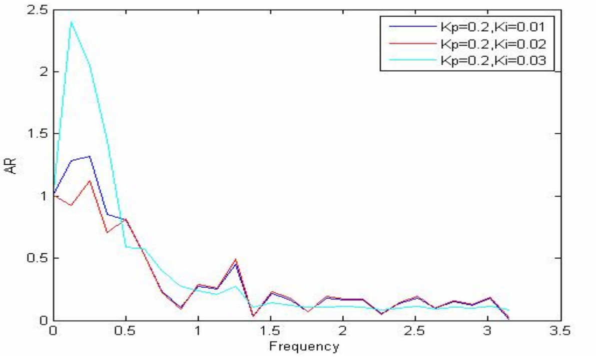


Figure .8 FR For The Pi Controller As Replenishment Rule.

5. REPLENISHMENT POLICIES VS PI CONTROL ACTIONS

In this section we will observe the mathematics structure of replenishment strategies approach by other authors Marko and Rusjan (2008), in order to classify them according to the laws of control, and the type of feedback that is used.

The policies analysed basically consists of three major components: exponentials smoothing of the demand (feedforward), order smoothing feedback loop and inventory position smoothing feedback loop.

These strategies are based on the demand forecasting and also on the feedback of signals as the level of inventory (I) and the previous orders $O_{i,i+1}$ (k-1). Our research highlights the most used strategies:

Strategy 1: We observe that usually other authors use the demand forecasting from downstream echelons through an exponential smoothing to estimate demand for the next period, that is:

$$O_{i,i+1}(k)=\hat{O}_{i-1,i}(k)=\hat{O}_{i-1,i}(k-1)+\alpha(O_{i-1,i}-\hat{O}_{i-1,i}(k-1)) \quad (28)$$

This policy reduces the bullwhip effect but does not ensure a safety inventory level; therefore this does not ensure the satisfaction of demand.

Observe that with the notation used, $O_{i-1,i}(k)$ represents the observed downstream orders (Demand) from the previous period, which we tried to predict by the demand forecast made in the previous period $(k-1)$, $\hat{O}_{i-1,i}(k-1)$.

The transfer function is give by:

$$\frac{O_{i,i+1}(z)}{O_{i-1,i}(z)}=\frac{\alpha z}{z+\alpha-1} \quad (29)$$

Strategy 2: A rule with a similar form that the exponential smoothing where order quantity $O_{i,i+1}(k-1)$ plays the role of orders from downstream (demand) forecasting $\hat{O}_{i-1,i}(k-1)$ is derived from the equation (16) which is given by:

$$O_{i,i+1}(z)=O_{i,i+1}(k-1)+\gamma\left(\hat{O}_{i-1,i}(k)-O_{i,i+1}(k-1)\right) \quad (30)$$

In this rule there exists an order smoothing feedback loop where the parameter γ has the same role as constant α in the equation (16) with $0<(\gamma)<1$.

Strategy 3: This rule is somewhat more complicated than the previous two, due to introduction concepts such as the inventory position, lead time and safety inventory

$$O_{i,i+1}(z)=\hat{O}_{i-1,i}(k)+\beta\left(I_0(z)-I_i\right) \quad (31)$$

This strategy is analogue to the approach in this paper since there exists a feedback level of inventory by a proportional factor 0<β<1 which coincides with our stability analysis. This policy allows control of inventory levels and reduces considerably the bullwhip effect.

We can see it is feasible to use different combinations in order to improve the reduction of bullwhip effect.

6. CONCLUSIONS

The discrete model for an echelon of the beer game has been derived using the z-transform. Some alternative ordering policies were formulated as P and PI control schemes. We obtain the characteristic equations of the closed loop and the stability of the system for asymptotic values has been investigated. The bullwhip effect is also analyzed through FFT over (AR), and we can conclude that using P and PI controllers the bullwhip effect of a supply chain unit can be suppressed. Finally we can see that the mathematical structure of replenishment strategies approach by other authors can be view as laws of control. We can therefore conclude that control theory is applicable to analysis of supply chains and that is would be possible to improve results using more efficient controllers as PI and PID.

REFERENCES

Dejonckheere, J., Disney, S.M., Lambrecht, M.R., Towill, D.R., 2002. Measuring and avoiding the bullwhip effect: A control theoretic approach. European Journal of Operating Research, 147, 567–590.

Dejonckheere, J., Disney, S.M., Lambrecht, M.R., Towill, D.R., (2004). The impact of information enrichment on the bullwhip effect in supply chains: a control engineering perspective. European Journal of Operating Research, 153, 727–750

Dragana, M.N, Panić, B, Vujošević, M. Bullwhip effect and supply chain modelling and analysis using cpn tools http://www.daimi.au.dk/CPnets/workshop04/cpn/papers/makajic-ikolic_panic_vujosevic.pdf [Accessed 15 September 2007].

Hoberg, K., James, R., Bradley, Ulrich, W., and Thonemann. (2007). Analyzing the effect of the inventory policy on order and inventory variability with linear control theory. European Journal of Operational Research, 176, 1620–1642.

Marko, J., Rusjan, B., (2008). The effect of replenishment policies on the bullwhip effect: A transfer function approach. European Journal of Operational Research, 184, 946–961.

Ogata, K., (1996). Sistemas de Control en Tiempo Discreto, México: Prentice Hall.

FUZZY SIMULATION FOR INFRASTRUCTURE EFFECTS UNCERTAINTY ANALYSIS

Donald Dudenhoeffer[(a)], Milos Manic[(b)]

[(a)]Idaho National Laboratory
[(b)]University of Idaho

[(a)]Donald.dudenhoeffer@inl.gov, [(b)] misko@uidaho.edu

ABSTRACT

In this paper we propose a method for conducting infrastructure effects-based modeling in uncertain environments. Critical infrastructure is composed of intertwining physical and social networks. Events in one network often cascade to other networks creating a domino effect. This cascading effect is not always well understood due to uncertainties in the multiple levels of effect. To account for these uncertainties, we present a method using fuzzy finite state machines (FFSM).

Keywords: fuzzy simulation, critical infrastructure, decision support system, interdependency modeling

1. INTRODUCTION AND MOTIVATION

Critical infrastructure in the United States is defined as “systems and assets, whether physical or virtual, so vital to the United States that the incapacity or destruction of such systems and assets would have a debilitating impact on security, national economic security, national public health or safety, or any combination of those matters. “ (U.S. Congress 2001) The categorization of such infrastructures varies slightly between countries, but is consistent in principle. The U.S. Government breaks the infrastructure into thirteen individual sectors:

- Agriculture;
- Food;
- Water;
- Public Health;
- Emergency Services;
- Government;
- Defense Industrial Base;
- Information and Telecommunications;
- Energy;
- Transportation;
- Banking and Finance;
- Chemical Industry; and
- Postal and Shipping. (Clinton 1996)

Sectors in turn contain individual infrastructures such as highways, rail systems, electric power generation and distribution, etc. Some of these systems are managed by government agencies, but the majority resides with industry. These infrastructures are characterized by a complexity of intertwined relationships that exist due to such factors as growing technological connectivity and economic requirements for distributed operations.

While this interconnectivity has increased information exchanged and improved efficiency of operations, it has also resulted in a potential chain of effect such that when a system is acted upon by an external force, it causes a domino effect or rippling of reaction not only within its own sector, but across multiple dimensions of infrastructure.

Effects Analysis, also referred to as effects-based and interdependency analysis, centers on gaining understanding on the resulting chain of effect that results when a system is perturbed by an event. While system dynamics may be well modeled and understood along individual infrastructures, such as in an electric power grid model or a water distribution model, how multiple infrastructures interact and affect each other, especially in light of upset conditions challenge current day understanding. Primary effects of an event are most often immediately observable and understandable; the subsequent chains of events that occur are less understood. The lack of appreciation for these second order, third order, n-order effects pose a serious problem for decision makers and responders in the event of global, national, and local event response.

A common method of representation for the systems of study in Effects Analysis is to use a directed graph or digraph and observe impact propagation. The system is decomposed into a set of key assets, and interdependent relationships, which in turn are modeled in a directed graph G(N,E) where N, the set of nodes, represents key assets, and E, the set of edges, represents the relationship between nodes. The system itself may represent functionally, physically, or behaviorally related group of regularly interacting or interdependent elements; that group of elements forming a unified whole.

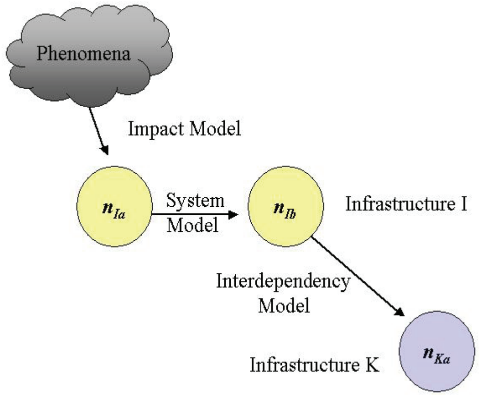


Figure 1: Chain of effects from phenomena to emergent impact

1.1. Uncertainty

To better understand these issues and representation, let n_{Ia} and n_{Ib} ∈ N be interconnected system nodes from infrastructure I, and the supplier of some resource or influence. Further let n_{Ka} ∈ N be a system node from infrastructure K and the consumer of that resource with edge $e(I_b,K_a)$, ∈ E representing the resource flow between the two nodes. Interest therefore arises when a phenomena or event disrupts infrastructure I, causing a cascading effect into infrastructure K as illustrated in Figure 1.

Such analysis, however, is complicated by uncertainty. This is truly a system of systems model and may incorporate various sub-models to represent the nature of the phenomena, the impact that the phenomena has on an infrastructure system, the infrastructure system model itself, and finally an interdependency model that captures the relationship between different infrastructures.

At each of these system layers or models, uncertainty can exist. Further, this uncertainty may be very difficult to identify and thus quantify. As an example consider the attempt to model the impact of hurricane wind damage on housing structures and predict the subsequent impact on not only the individual structures, but on electrical power, and on population mobility trends during and immediately after the impact event.

Wind damage models may be employed to estimate the potential damage to physical structures. This information can be represented in the form of a damage matrix, Table 1, or a damage curve, Figure 3. This damage matrix developed by Filliben et. Alt. (2002) list the probability of damage to a residential structure broken down by loss or roof and by total

Table 1. Damage Matrix for Residential Housing

v mph		90	100	110	120	130	140
R	P(R/v)	0.03	0.2	0.5	0.8	1.0	1.0
C	P(C/v)	0	0	0.02	0.05	0.2	1.0

Several issues arise from using this information in an infrastructure effects model.

1. Damage predictions are usually based on models utilizing either generic specifications or on the actual infrastructure in a specific area. In the first case, damage predictions on basic structural requirements do not account for the specifics of geographical areas. Conversely, in the second case, modeling for specific geographical areas, does not necessarily capture features outside of that area.
2. While absolutes can be modeled with high confidence, intermediates states are more difficult to ascertain.

Additionally, the primary effect may not be the main issue of concern, but a subsequent emergent behavior might be the focus of analysis. Consider the case of wind damage in a residential community. A driving concern to emergency coordinators and responders, may not be the damage itself, but the mobility patterns of the residents and the effect that changes in infrastructure plays, i.e. when do residents vacate and return? In this case, wind damage may be only one consideration to residential mobility. Other factors may include electric power, proximity to the event, availability of transport, etc…

As such, one can see that effects analysis modeling is wrought with multiple levels of uncertainty. The goal of effects analyses should not be the determination of a precise outcome, but it should be to identify a set of possible outcomes to a given event or series of events. It is with this consideration that we introduce fuzzy set theory and fuzzy simulation as a means to model sets of possible emergent effects.

2. FUZZY SIMULATION

2.1. Uncertainty in Simulation Models

Computer simulation is the attempt to gain understanding of real work phenomena that are too complex for strictly analytical evaluation. A computer simulation uses a mathematical model to represent the object system, initial conditions (i.e. an initial data sets and assumptions on system state) are established, and one or more iterations of the simulation are run to gain understanding on system performance.

While the use of computer simulation continues to grow, it is important to understand the issues and limitations associated with its use. Specifically, some potential disadvantages include:

- A stochastic simulation only produces an estimate of a systems true characteristics based on a set of initial inputs (Law and Kelton 1982)

- It is possible to place too much confidence in the result of a simulation's outcome. If either the underlying model or the data is incorrect, then the results of the simulation will reflect a level of uncertainty. (Law and Kelton 1982)
- The demand for numerical precision and measurability may lead to over simplification and approximation also introducing uncertainty (Anglani 1998)

Aló et alt (2002) further emphasizes the above points in stating that in modeling decision making under uncertainty, it is often the case that the decision maker does not know or have full understanding of the true "state of the world" surrounding his/her decisions. Further, this lack of knowledge may also include the lack of probabilistic data associated with the potential states of the world. These limitations, however, do not diminish the value that computer simulation adds in system understanding. The key then is to both comprehend and mitigate these potential drawbacks. This paper focuses on a method to capture the multiple levels of uncertainties associated with infrastructure effects modeling.

Current simulation techniques to mitigate uncertainty in modeling and simulation results include the use of confidence bounds on simulation results. Multiple methods can be employed to establish and minimize these uncertainty bounds including conducting large numbers of stochastic simulation runs, analyzing input data precision, and incorporating model uncertainty into the final output. The method that we propose for infrastructure effects modeling is the use of fuzzy set theory not only to capture uncertainty, but to preserve the nature of the uncertainty throughout the simulation process.

2.2. Fuzzy Simulation Integration

The integration of fuzzy set theory into simulation has been proposed and demonstrated in earnest since the mid-1990's. Anglani et alt (1998) used fuzzy sets by to model the uncertainty of the time interval between events in a discrete simulation. He states that the integration of fuzzy sets into simulations allows the formulation and solution of problems whose complexity or simply the lack of state knowledge inhibits the use of traditional mathematical models in solution development.

Sevastjanov and Rog (2003) likewise used fuzzy sets to model the interval between events over the more traditional approach of a probabilistic distribution of times for events in a logistics simulation. Hullermeier (1996) citing that the knowledge of dynamical systems is often vague or ill defined, applied fuzzy set principles in developing a differential equation model for the prediction of object trajectory in spatiotemporal reasoning. Still, however, the application of fuzzy principles has not been readily adopted in all simulation application areas.

To support infrastructure effects analysis, we propose the application of fuzzy finite state machines (FFSM) to model asset state in infrastructure modeling. At the simplest level a finite state machine can be described as "a collection of inputs, a collection of outputs, and a finite collection of states, which describe the effect of the various inputs signals." (Wilson and Watkins 1990). More precisely given the current state and a set of inputs, the finite state machine, based upon a defined rule set, determines the next state, and maps input signals to output signals. A FFSM is the implementation of the principles of a finite state machine, but allows the system to deal with the reality of non-precise or non-crisp sets of state, inputs, and outputs. One of the first descriptions of FFSM's implementation was by Grantner and Patyra (1993) which described the used of fuzzy logic state machines in VLSI implementation. A more recent work by Grantner et alt (2000) described the use of FFSM in addressing ontological control problems and recovery actions in large PC-based systems.

3. MODEL DEVELOPMENT

As discussed in Section 1, one issue in modeling phenomena impact and the subsequent effects is that the impact that a trigger event imparts on an infrastructure item may not be precisely known in terms of both immediate and lasting influence. Multiple methods exist to capture this uncertainty. One method may be to use a probabilistic distribution function model using the expected (i.e. average) effect or the worst-case scenario in modeling. The method that we propose, however, is the incorporation of fuzzy sets to capture the uncertainty of effect for individual events and further, to utilize this concept to carry forward uncertainty as effects cascade forward in time.

In addition to uncertainty associated with the effect of the trigger event, uncertainty may also exist as to the exact state of the entity or node in questions. Due to incomplete knowledge or immeasurable status, some states may not be fully understood prior to the need to model them. Examples include the physical status of a piece of equipment or facility that is neither under the direct control nor immediately observable by the modeler. Also consider the example of more subjective nodes such as public confidence or public opinion concerning particular topics. Aló et alt. (2002) discuss similar issues in while applying fuzzy functions to derive optimal decisions in uncertain environments.

The approach that we have taken is to define a node as the tuple: N(I, E, Sp , Sn, O, Fe, Fp) where

I = a nonempty finite set of input entities required for node operation
E = a nonempty finite set of trigger events
S_p = a nonempty finite set of present states
S_n = a nonempty finite set of next states
O = a nonempty finite set of output entities resulting from node operation (F_p)

F_e = the mapping of effect of a trigger event on the current node state S_p
F_p = is the node process associated with the transformation of input (I) to output (O) (i.e. State transition function).

Figure 2 illustrates this concept.

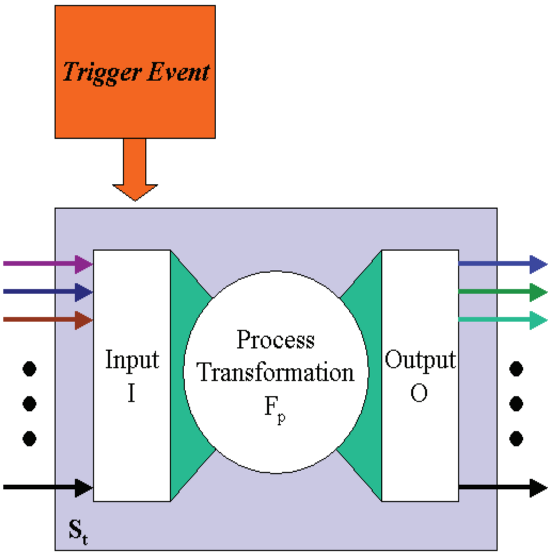


Figure 2. Node functional representation.

Consider the case where the effect of an event is not precisely known. We will model this uncertainty in the form of a fuzzy set and fuzzy mapping of consequence.

Let S = {s_1, s_2, s_3, …s_n} be the possible state of node N with Sp ∈ S and S_n ∈ S. Let E = {e_1, e_2, e_3, … e_m} represent possible trigger events. High-level examples of trigger events may include fire, flooding, a tornado, component failure, human error, and malicious attack.

As defined earlier, F_e is the impact relation that a trigger event will have on the present state, S_p. Thus F_e: S x E → S. Note that the impact of an event depends not only on the nature of the event, but may also depend on the current state of the system. Under perfect conditions, this relationship would be precisely defined. In reality however, uncertainty often exists as the to exact nature of the effect. To account for uncertainty, define Ω_e as the set of fuzzy of potential outcomes (i.e. new states) that could result from an effect e. Further define μ as the membership function of S in Ω_e,

$$\text{s.t.} \quad \mu(s) = \text{ for s } \Omega \qquad (1)$$

$$\mu(s) \in [0,1] \text{ for s } \Omega \qquad (2)$$

$$\text{So } \Omega = \{s_1/\mu(s_1), s_2/\mu(s_2), \ldots\ldots. s_n/\mu(s_n)\}. \qquad (3)$$

Membership values of μ(si) = 0 are not included in the set for brevity. Now the F_e function becomes a fuzzy mapping, F_e: S x E → Ω_e where Ω_e represents the uncertainty of effect for event e. Consider the following example. The node (i.e. asset of interest) is an electrical substation described by the following tuple: N(I, E, Sp , Sn, O, Fe, Fp) with

I = {0 (no electricity), 1(electricity)};
S = {0 (shutdown, requiring repairs),
1 (shutdown, no Input power),
2 (operational)};
O = {0 (no electricity), 1 (electricity)};
E = {0 (flooding to a level > 3 feet),
1 (fire in the substation),
2 (equipment failure),
3 (malicious attack),
4 (system repair)}

Fp is given in Table 1.

Table 2. State Transition Matrix

S_p	Input	S_n	Output
0	0	0	0
0	1	0	0
1	0	1	0
1	1	2	1
2	0	1	0
2	1	2	1

Consider a flooding event occurs that results in localized flooding to a depth of 3 feet at the substation of concern. Based upon damage analysis, the following impact matrix represents the state transitions that could occur over all possible initial states.

Table 3. Impact Matrix

Ω_e =

	$S_n = 0$	$S_n = 1$	$S_n = 2$
$S_p = 0$	1	0	0
$S_p = 1$	.8	.5	0
$S_p = 2$	.7	.5	.3

The rows represent the node's initial state and the columns represent the end state as a result of the event. The cells represent the membership value μ(s) for the combination (s_i, s_j). Further let the initial state be denoted by the following state matrix S_p = [0 1 0] to represents the level of membership in the current state. The end state membership of the event then is denoted by $S_n = S_p \times \Omega_e$ = [.8 .5 0].

While the end state of the substation is important, it is also important to examine this impact on other nodes within the interdependent network. Another aspect to evaluate is the effect on node output. Consider the mapping from S_n to O as described by O(s) = $[0\ 0\ 1]^T$. Then the uncertainty of output can be described by the following fuzzy set Ωo. Here Ωo =

[0/.7 0/.5 1/.3] which can be reduced to Ωo = [0/.7 1/.3].

The values of Ωo subsequently provide input to another node along the network and hence uncertainty is propagated down the system path.

4. PRELIMINARY RESULTS

As a proof of concept for the integration of fuzzy simulation into effects analysis, we constructed an infrastructure simulation utilizing fuzzy finite state machines. The framework we used for integrating the FSSM was CIMS© (Dudenhoeffer et alt 2006).

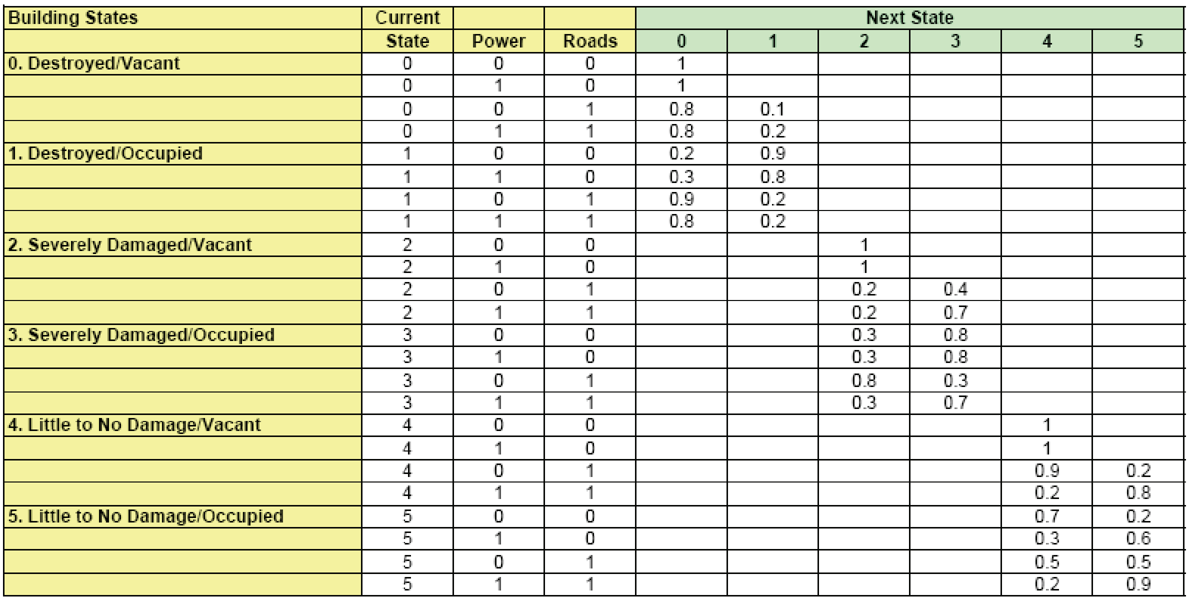
CIMS©, the Critical Infrastructure Modeling System, was developed at the Idaho National Laboratory to examine the interrelationships between infrastructure networks and more specifically, the emergent systems behaviors that develop when one or more nodes within the system are perturbed. A discrete event simulation, CIMS© uses an agent-based approach (ABM) (Rocha 1999) to model infrastructure elements, the relations between elements, and individual component behavior. The key characteristic of the agent and the simulations is that each agent exists as an individual entity which maintains a state, senses input, and possesses rules of behavior that act upon the inputs and either modify the state or produce an output. Each network within the simulation is modeled as a connected graph, G = (N, E), where N represents the nodes within the network and E represents the edges between the nodes. Edges also represent the relationship, i.e. interdependencies, between infrastructures.

The modification to CIMS© involved replacing deterministic process functions with a fuzzy transition table and a fuzzy state matrix. Additionally, where as before, input and output flow were crisp quantities, now they support passing fuzzy sets between nodes as input and output. Finally, CIMS© supports the insertion of events and the creation of event driven scenarios. This was modified to account for fuzzy effects from events. A nice feature of the FFSM integration with the CIMS© software package is that it allows the simultaneous modeling of both crisp and fuzzy relations.

The state transition algorithm used the Max-Min Composition relation (Tsoukalas and Uhrig 1997) to calculate the next state, as illustrated

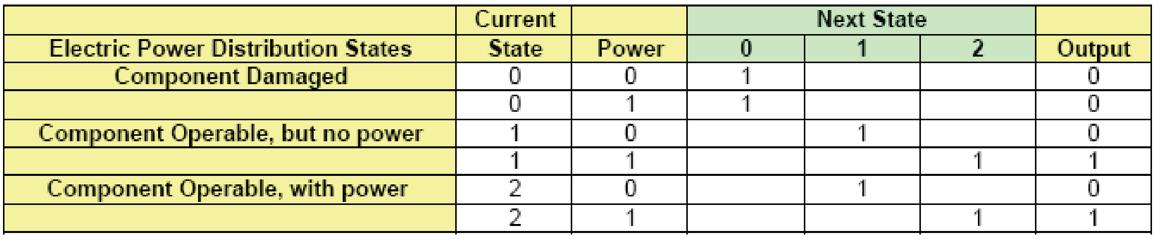
$$R_1 \circ R_2 \equiv \int_{X\,x\,Z} \underset{y}{\vee} \left[\mu_{R_1}(x,y) \wedge \mu_{R_2}(x,z)\right]/(x,z) \qquad (4)$$

where R_1 is the initial system state including the current input matrix and R_2 is the state transition matrix. The resulting matrix is the next state matrix, which is then used to calculate the node's output.



Building States	Current			Next State					
	State	Power	Roads	0	1	2	3	4	5
0. Destroyed/Vacant	0	0	0	1					
	0	1	0	1					
	0	0	1	0.8	0.1				
	0	1	1	0.8	0.2				
1. Destroyed/Occupied	1	0	0	0.2	0.9				
	1	1	0	0.3	0.8				
	1	0	1	0.9	0.2				
	1	1	1	0.8	0.2				
2. Severely Damaged/Vacant	2	0	0			1			
	2	1	0			1			
	2	0	1			0.2	0.4		
	2	1	1			0.2	0.7		
3. Severely Damaged/Occupied	3	0	0			0.3	0.8		
	3	1	0			0.3	0.8		
	3	0	1			0.8	0.3		
	3	1	1			0.3	0.7		
4. Little to No Damage/Vacant	4	0	0					1	
	4	1	0					1	
	4	0	1					0.9	0.2
	4	1	1					0.2	0.8
5. Little to No Damage/Occupied	5	0	0					0.7	0.2
	5	1	0					0.3	0.6
	5	0	1					0.5	0.5
	5	1	1					0.2	0.9

Figure 3: Building/Facility Fuzzy State Transition Table



	Current		Next State			
Electric Power Distribution States	State	Power	0	1	2	Output
Component Damaged	0	0	1			0
	0	1	1			0
Component Operable, but no power	1	0		1		0
	1	1			1	1
Component Operable, with power	2	0		1		0
	2	1			1	1

Figure 4: Power Distribution Substation Fuzzy State Transition Table

The preliminary model developed to demonstrate the application of FFSM's in the simulation involves an evaluation of resident mobility given in a storm damage situation. Specifically we wish to evaluate the behavior of residents on evacuating their primarily residence. The work presented here does not reflect actual data to this point, but reflects a potential framework for evaluating this situation.

Given the area or interest, the first step was to model resident behavior in the form of discrete states. Here five states were identified for purposes of the simulation. Potential factors contributing to the decisions on occupancy that we incorporated into the model included electrical power and accessibility (i.e. roadway passage) to and from the residences. Figure 3 provides the transition matrix (R_2) used to determine the next state. The next states are represented in the table by their membership values. For example the transition from state 5 (Little or No Damage to residence, occupancy is maintained) given that Power is available and Roads are available $S_p \rightarrow S_n = \{4/0.2, 5/0.9\}$. Figure 4 represents the fuzzy state transition table for the electric substations.

4.1. Test Case

The simulation scenario centers on the impact of hurricane like winds in an urban setting. Figure 5 displays the subset of interest, which has been modeled using FFSM's in CIMS©.

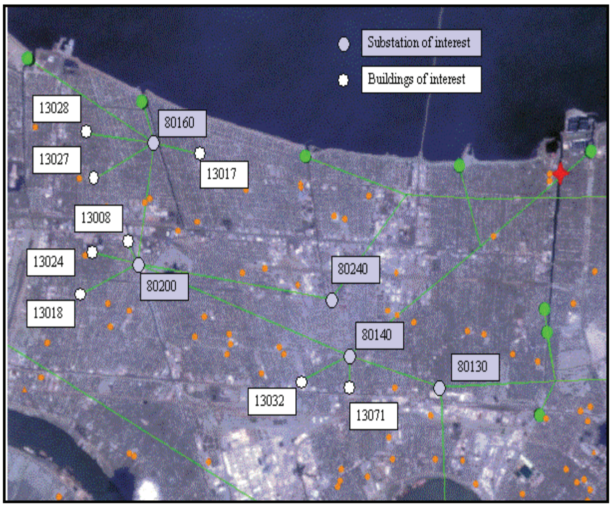


Figure 5: CIMS© network of fuzzy nodes and edges.

The primary nodes of interest are identified by their ID numbers in the figure and represent key facilities and electrical power substations. The green edges between then nodes represent the electric power supply to the facilities. The CIMS© framework with the fuzzy nodes and relationships allows the user to examine the propagation of uncertainty along system dependent relationships. Prior to this added feature, results were presented as crisp state output with no uncertainty representation.

Table 4. Run A – Base Case

```
80140  Substation   Labarre
   state: 0  value:    0
   state: 1  value:    0
   state: 2  value:    1
   Input power off:    0
   Input power  on:    1
   NEXT STATE
   state: 0  value:    0
   state: 1  value:    0
   state: 2  value:    1
   OUTPUT
   Power Off:    0
   Power On :    1
```

```
13071  Saint Christopher School
   state: 0  value:    0
   state: 1  value:    0
   state: 2  value:    0
   state: 3  value:    0
   state: 4  value:   0.2
   state: 5  value:   0.8
   Input power off:    0
   Input power  on:    1
   NEXT STATE
   state: 0  value:    0
   state: 1  value:    0
   state: 2  value:    0
   state: 3  value:    0
   state: 4  value:   0.2
   state: 5  value:   0.8
```

Table 5. Run B – Fuzzy Impact Event

```
80140  Substation   Labarre
   state: 0  value:   0.7
   state: 1  value:    0
   state: 2  value:   0.4
   Input power off:    0
   Input power  on:    1
   NEXT STATE
   state: 0  value:   0.7
   state: 1  value:    0
   state: 2  value:   0.4
   OUTPUT
   Power Off:   0.7
   Power On :   0.4
```

```
13071  Saint Christopher School
   state: 0  value:    0
   state: 1  value:    0
   state: 2  value:    0
   state: 3  value:    0
   state: 4  value:   0.5
   state: 5  value:   0.5
   Input power off:   0.7
   Input power  on:   0.4
   NEXT STATE
   state: 0  value:    0
   state: 1  value:    0
   state: 2  value:    0
   state: 3  value:    0
   state: 4  value:   0.5
   state: 5  value:   0.5
```

The first simulation run, Run A, is the base case, which shows all substations operating without event or disturbance. The purpose of this run was just to show the program consistency. Here Table 4 shows the results for two nodes, substation 80140 and building 13071.

The next run, Run B, illustrates the insertion of a fuzzy event at Substation 80140 which alters its state to $S_n = \{0/0.7, 1/0.4, 2/0.2\}$ and examines the propagation of the uncertainty of effect forward. The resulting impact on the substation and the cascading impact on Node 13071 is shown in Table 5.

5. CONCLUSION

In this paper we have discussed the research area of effects analysis and the challenges in modeling the uncertainty associated with unknown or imprecise cause-effect relationships. As one possible modeling tool, we demonstrated the application of fuzzy finite state machines to capture and propagate uncertainty across multiple effects. This principle was demonstrated in a simulation package called CIMS©. Preliminary results show that this has potential in providing decision makers with a means for better understanding the uncertainty and possible cause-effect paths resulting from infrastructure events.

REFERENCES

United States Congress. 2001. U.S.A. Patriot Act [online]. Available via < http://www.epic.org/privacy/terrorism/hr3162.html > [accessed Mar 21, 2006]

Clinton, W. J. 1996. Executive Order 13010—Critical Infrastructure Protection. Federal Register, 1996. Vol. 61, No. 138: 37347-37350.

Filliben, J.J., K. Gurley, J.P. Pinelli and E. Simiu. 2002. Fragility curves, damage matrices, and wind induced loss estimation, In *Proceedings from the Third International Conference on Computer Simulation in Risk Analysis and Hazard Mitigation*, June 19-21, 2002, Sintra, Portugal, 119-126 pp.

Law, A.M. and W.D. Kelton, 1982. *Simulation Modeling & Analysis*, New York, McGraw-Hill, Inc.

Alo, R. de Korvin, A. Modave, F. 2002. Using Fuzzy functions to select an optimal action in decision theory. In *Fuzzy Information Processing Society, 2002. Proceedings. NAFIPS. 2002 Annual Meeting of the North American*, pp.348-353.

Anglani A., Grieco A., Nucci F., Semeraro Q., Tolio T. 1998. Representation and use of uncertainty in discrete event simulation models, In *Proceedings of the 10TH European Simulation Symposium and Exhibition - Simulation in Industry*, Oct. 26-28 - Nottingham, U. K.

Hullermeier, E. 1996. A fuzzy simulation method. In P.G. Anderson and K. Warwick, editors, *International Symposium on Soft Computing*, pages B230–B236, Reading, U.K., ICSC Academic Press.

Wilson, R.J and J. Watkins. 1990. *Graphs: An Introductory Approach: A First Course in Discrete Mathematics*. Wiley, p.100.

Grantner, J., M. Patyra, 1993. VLSI implementation of fuzzy logic finite state machines, *IFSA ′93 World Congress, Proceedings* Vol. II, pp. 781-784, Seoul, Korea.

Grantier, J., G.A. Fodor, D. Driankov, and M.J. Patyra. 2000. Applications of the Fuzzy State Fuzzy Output Finite State Machine to the Problem of Recovery from Violation of Ontological Assumptions, International Journal of Smart Engineering System Design, Vol, 2, No. 3, pp. 177-199.

Rocha, L. M. 1999. Complex System Modeling: Using Metaphors from Nature in Simulation and Scientific Methods. BITS: Computer and Communications News. Los Alamos National Laboratory, November 1999. Available via <http://www.c3.lanl.gov/~rocha/complex/csm.html> [accessed March 26, 2002].

Dudenhoeffer, D., M. Permann, and M. Manic, 2006. CIMS: A Framework for Infrastructure Interdependency Modeling and Analysis, *In Proceedings of the 2006 Winter Simulation Conference*, IEEE, December 2006, pp 478--485.

Tsoukalas, L.H. and R.E. Uhrig. 1997. *Fuzzy and neural approaches in engineering*, John Wiley & Sons, New York: 1997

AUTHORS BIOGRAPHY

DONALD D. DUDENHOEFFER is the Senior Vice President of Priority 5 Holding, Inc., a critical infrastructure management solutions company. Prior to that, he was Department Manager at the Idaho National Laboratory (INL) in Idaho Falls, ID. He received his Masters of Science degree in Operations Research from the Naval Postgraduate School in 1994. His research interests include critical infrastructure modeling and simulation, military operations, command and control.

MILOS MANIC is the Graduate and Undergraduate Program Coordinator for the CS & ECE program, Center for Higher Education of University of Idaho at Idaho Falls, and an Assistant Professor at the Computer Science Dept., teaching graduate courses and workshops. He holds a Ph.D. in Computer Science from University of Idaho Boise and a Masters in Electronic Engineering and Computer Science from University of Nis. His research interests include artificial intelligence, decision support systems, and software reliability.

A PARALLEL PROGRAMMING METHODOLOGY USING COMMUNICATION PATTERNS NAMED CPANS OR COMPOSITION OF PARALLEL OBJECT

M. Rossainz-López[(a)], M. I. Capel-Tuñón[(b)]

[(a)] Universidad Autónoma de Puebla, Avenida. San Claudio y 14 Sur, San Manuel, Puebla,
State of Puebla, 72000, México
[(b)] Departamento de Lenguajes y Sistemas Informáticos, ETS Ingeniería Informática,
Universidad de Granada, Periodista Daniel Saucedo Aranda s/n,
18071 Granada, Spain

[(a)]mariorl@siu.buap.mx, [(b)]mcapel@ugr.es

ABSTRACT

Within an environment of Parallel Objects, an approach of Structured Parallel Programming and the paradigm of the Orientation to Objects, shows a programming method based on High Level Parallel Compositions or HLPCs (CPANs in Spanish) by means of classes. The synchronous, asynchronous communication ways and asynchronous future of the pattern of Parallel Objects (Rossainz and Capel 2005-2), the predefined patterns of communication/interaction of the structured approach, the encapsulation and the abstraction of the Orientation to Objects, to provide reusability to this patterns, together with a set of predefined restrictions of synchronization among processes (maxpar, mutex, sync) are used. The implementation of the commonly used communication patterns is explained, by means of the application of the method, which conform a library of susceptible classes of being used in applications within the environment of programming of the C++ and of the standard POSIX of programming with threads.

Keywords: Parallel Objects, Structured Parallel Programming, High Performance Computing, Object Oriented Programming.

1. INTRODUCTION

As it is known, exist infinity of applications that using machines with a single processor tries to obtain the maximum performance from a system when solving a problem; however, when such a system can not provide the performance that is required (Capel and Troya 1994), a possible solution it consists on opting for applications, architectures and structures of parallel or concurrent processing. The parallel processing is therefore, an alternative to the sequential processing when the limit of performance of a system is reached. In the sequential computation a processor only carries out at the same time an operation, on the contrary of what happens in the calculation parallel, where several processors they can cooperate to solve a given problem, which reduces the time of calculation since several operations can be carried out simultaneously. From the practical point of view, today in day is enough justified carrying out compatible investigations within the area of the parallel processing and areas related (Concurrence, Distributed Systems, Systems of Real Time, etc.), since the recent advance in massively parallel systems, communications of great band width, quick processors for the treatment of signs, etc., they allow this way it. Important part of those investigations are the parallel algorithms, methodologies and models of parallel programming that at the moment are developing. The parallel processing includes many topics that include to the architectures, algorithms, languages of programming parallel and different methods of performance analysis, to mention some of the most excellent.

The present investigation centers its attention in the Methods of Structured Parallel Programming, proposing a new implementation with C++ and the library of threads POSIX of the programming method based on the pattern of the High Level Parallel Compositions or CPANs (Corradi 1995; Danelutto), the which it is based on the paradigm of Orientation to Objects to solve problems parallelizable using a class of concurrent active objects. In this work supply a library of classes that provides the programmer the communication/interaction patterns more commonly used in the parallel programming, in particular, the pattern of the pipeline, the pattern denominated farm and the pattern tree of the technique Divide and Conquer of design of algorithms, well-known as it.

2. MOTIVATION

At the moment the construction of concurrent and parallel systems has less conditioners every time, since the existence of systems parallel computation of high performance, or HPC (High Performance Computing), more and more affordable, has made possible obtaining a great efficiency in the processing of data without the cost is shot; even this way, open problems that motivate the investigation in this area still exist; in particular, interest us those that have to do with the parallel applications that use communication patterns predetermined among their component software. At the moment are identified as important open problems, at least, the following ones:

The lack of acceptance of environments of parallel programming structured to develop applications: the structured parallelism is a type of parallel programming based on the use of communication/interaction patterns (pipelines, farms, trees, etc.) predefined among the processes of user's application. The patterns also encapsulate the parallel parts of this application, of such form that the user only programs the sequential code of this one. Many proposals of environments exist for the development of applications and structured parallel programs, but until the moment, they are only used by a very limited circle of expert programmers. At the moment, in HPC, a great interest exists in the investigation of environments as those previously mentioned ones.

The necessity to have patterns or high level parallel compositions: a high level parallel composition or CPAN, as well as is denominated in (Corradi 1995), it must be been able to define and to use within an infrastructure (language or environment[1] of programming) oriented to objects. The components of a parallel application not interaction in an arbitrary way, but regular basic patterns follow (Hartley 1998). An environment of parallel programming must offer its users a set of components that implement the patterns or CPANs more used in algorithms and parallel and distributed applications, such as trees, farms, pipes, etc. The user, in turn, must can to compose and to nest CPANs to develop programs and applications. The user must be limited to a set of predefined CPANs, but rather, by means of the use of the inheritance mechanism, he must can to adapt them to his necessities. The development environment must contemplate, therefore, the concept of class of parallel objects. Interest exists in exploring the investigation line related with the definition of complete sets of patterns, as well as in its semantic definition, for concrete classes of parallel applications.

Determination of a complete set of patterns as well as of their semantics: in this point, the scientific community doesn't seem to accept in a completely satisfactory way and with the enough generality none of the solutions that have been obtained to solve this problem today. It doesn't seem, therefore, easy the one that can be found a set the sufficiently useful and general thing, for example a library of patterns or set of constructos of a programming language, to be used in the development, in a structured way, of a parallel application not specific.

Adoption of a approach oriented to objects: Integrating a set of classes within an infrastructure oriented to objects is a possible solution to the problem described in the previous point, since would allow adding new patterns to an incomplete initial set by means of the subclasses definition. Therefore, one of the lines of followed investigation has been finding representations of parallel patterns as classes, starting from which instance parallel objects is been able to (CPANs) that are, in turn, executed as consequence from an external petition of service to this objects and coming from user's application. For example, the derived pattern of the execution for stages of the processes would come defined by pattern's denominated pipeline class; the number of stages and the sequential code of each specific stage would not be established until the creation of a parallel object of this class; the data to process and the results would be obtained of user's application; the internal storage in the stages could adapt in a subclass that inherits of pipeline. Several advantages are obtained when following a approach oriented to objects (Corradi and Leonardi 1991), regarding a approach only based on skeletons algorithmic and programs model (Hartley 1998), it is necessary to point out, for example, the following improvements:

Uniformity: All the entities within the programming environment are objects.

Genericity: The capacity to generate references dynamically, within an environment of software development oriented to objects, makes possible the creation of generic patterns, by means of the definition of its components as generic references to objects. Reusability: The inheritance mechanism simplifies the definition of specialized parallel patterns. The inheritance applied to the behavior of a concurrent object helps in the specification of the parallel behavior of a pattern.

3. EXPOSITION OF THE PROBLEM

From the work carried out to obtain the investigating sufficiency presented in Julio 1999, redefining and modernizing the investigation, the problem to solve is defining a Parallel Programming Method based on High Level Parallel Compositions (CPANS) (Corradi 1995). For it the following properties have considered as indispensable requirements that should be kept in mind for the good development of this investigation. It is required, in principle, a environment of Programming Oriented to Objects that it provides: Capacity of invocation of methods of the objects that contemplates the asynchronous communication ways and asynchronous future. The asynchronous way doesn't force to wait the client's result that invokes a method of an object. The asynchronous future communication way makes the client to wait only when needs the result in a future instant of her execution. Both communication ways allow a client to continue being executed concurrently with the execution of the method (parallelism inter-objects).

---

[1] One talks about the concept of *HPC programming environment* : environment of "friendly" parallel programming to users that provide facilities for the development of applications, abstracting details of low level as the referred ones to the creation, allocation, coordination and communication of the processes in a distributed and parallel system.

The objects must can to have internal parallelism. A mechanism of threads it must allow to an object to serve several invocations of their methods concurrently (parallelism intra-objects).

Availability of synchronization mechanisms when parallel petitions of service take place. It is necessary so that the objects can negotiate several execution flows concurrently and, at the same time, to guarantee the consistency of their data.

Availability of flexible mechanisms of control of types. The capacity must be had of associating types dynamically to the parameters of the methods of the objects. It is needed that the system can negotiate types of generic data, since the CPANs only defines the parallel part of an interaction pattern, therefore, they must can to adapt to the different classes of possible components of the pattern.

Transparency of distribution of parallel applications. It must provide the transport of the applications from a system centralized to a distributed system without the user's code is affected. The classes must maintain their properties, independently of the environment of execution of the objects of the applications.

Performance. This is always the most important parameter to consider when one makes a new proposal of development environment for parallel applications. A approach based on patterns as classes and parallel objects must solve the denominated problem PPP (Programmability, Portability, Performance) so that it is considered an excellent approach to the search of solutions to the outlined problems.

The environment of programming oriented to Objects that it has been considered as suitable to cover the 6 previously mentioned properties is the programming language C++, together with the use of the standard POSIX Thread, having as base the operating system Linux, in particular the system Red Hat 7.0.

4. SCIENTIFIC OBJETIVES OF INTEREST

The development of a programming method is based on High Level Parallel Compositions or CPANs that implement a library of classes of utility in the Programming Concurrent/Parallel Oriented to Objects (Rossainz 2005). The method must provide the programmer the commonly used parallel patterns of communication, in such a way that this can exploit the generalization mechanisms for inheritance and parametrization to define new patterns according to the pattern of the CPAN. The specific objectives to reach in this work are:

- To develop a programming method based on High Level Parallel Compositions or CPANs
- To develop a library of classes of Parallel Objects (Rossainz and Capel 2005-2) that provides the user the patterns (under the pattern of the CPAN) more commonly used for the parallel programming.
- To offer this library to the programmer so that, with minimum knowledge of Parallelism and Concurrence, it can exploit them, by means of the use of different reusability mechanisms, under the paradigm of the Orientation to Objects and she/he can, also, to define own patterns, adapted to the communication structure among the processes of their applications.
- Transform known algorithms that solve sequential problems (and that they can be easily parallelizable) in their version parallel/concurrent to prove the methodology and the component software developed work presently.

5. HIGH LEVEL PARALLEL COMPOSITIONS OR CPANS

Some of the problems of the environments of parallel programming it is that of their acceptance for the users, which depends that they can offer complete expressions of the behavior of the parallel programs that are built with this environments (Corradi 1995). At the moment in the systems oriented to objects, the programming environments based on parallel objects are only known by the scientific community dedicated to the study of the Concurrence.

A first approach that tries to attack this problem it is to try to make the user to develop his programs according to a style of sequential programming and, helped of a system or specific environment, this can produce his parallel tally. However, intrinsic implementation difficulties exist to the definition of the formal semantics of the programming languages that impede the automatic parallelization without the user's participation, for what the problem of generating parallelism in an automatic way for a general application continues being open.

A promising approach alternative that is the one that is adopted in the present investigation to reach the outlined objectives, is the denominated structured parallelism. In general the parallel applications follow predetermined patterns of execution. These communication patterns are rarely arbitrary and not structured in their logic (Brinch Hansen 1993). The High Level Parallel Compositions or CPANs are patterns parallel defined and logically structured that, once identified in terms of their components and of their communication, they can be taken to the practice and to be available as abstractions of high level in the user's applications within an environment or programming environment, in this case the one of the orientation to objects. The structures of interconnection of more common processors as the pipelines, the farms and the trees can be built using CPANs, within the environment of work of the Parallel Objects that is the one used to detail the structure of the implementation of a CPAN.

5.1. The Structured Parallelism.

A approach structured for the parallel programming is based on the use of communication/interaction patterns (pipelines, farms, trees, etc.) predefined among the processes of user's application. In such a situation, the approach of the structured parallelism provides the interaction pattern's abstraction and it describes applications through CPANs able to already implement the patterns mentioned.

The encapsulation of a CPAN should follow the modularity principle and it should provide a base to obtain an effective reusability of the parallel behavior that is implemented. When it is possible to make this, a generic parallel pattern is made, which provides a possible implementation of the interaction among the processes, independent of the functionality of these.

The approach structured for the parallel programming in the last years has followed two ways basically:

1. The enrichment of traditional parallel environments with libraries of “skeletons” (Darlington 1999) of programs that concrete communication patterns represent.
2. The definition of restrictive and closed parallel languages that provide communication in terms of the patterns that are already defined in the system (Bacci 1999).

The approach presented here assists to the first way to consider it more generic and more open. What thinks about now is that, instead of programming a concurrent application from the beginning and of controlling the creation of the processes so much as that of the communications among them, the user simply identifies the CPANs that implement the patterns adapted for the necessities of communication of his application and it uses with the sequential code that implements the computations that individually carry out their processes.

They can be identified of way informal several significant parallel patterns of interconnection and reusable in multiple applications and parallel algorithms, but an agreement doesn't exist the sufficiently general thing that allows to define its semantic ones formally (Corradi 1995). For example, the patron farm is a concept that can be understood by most of its general possible users of a formal one, but its concretion in a particular application she/he forces these to choose among different strategies for its implementation.

5.2. The Object-Oriented.

Sometimes the consent lack in the semantics of the parallel patterns makes that its definition is usually complex and that this is only given at a low level of its implementation; therefore, it is forced the users to go into details of the architecture of the system when they are tried to use the patterns in a concrete program. However, in an environment of development of expandable software, as it is it the one from the object-oriented, the programmer can end up defining any parallel pattern that needs, via a language of high level or graphic tool that it supports the paradigm, adapting it, later on to the characteristics of a concrete application by means of the inheritance mechanisms and genericity.

The basic characteristic of these systems is the definition of independent modules of the context that can be connected to each other via channels of communication of high level. The obtaining of parallel compositions represents communication patterns then statically certain and that they can be built independently of the context and in modules reusable, providing this way the encapsulation of the parallel behavior and the capacity of anidation of modules. The basic idea is to define to the CPANs as objects in charge of to control and to coordinate the execution of its components interns. Under this premise you can create an environment of expandable development, based on CPANs that provides characteristic as important as they can be: uniformity, generality and reusability[2].

5.3. Definition of the pattern CPAN.

The basic idea is the one of implementing any type of parallel patterns of communication between the processes of an application or distributed/parallel algorithm as classes, following the paradigm from the Orientation to Objects. Starting from this classes, an object can be instanced and the execution of a method of the object in question you can carry out through a petition of service. A CPAN comes from the composition of a set of objects of three types (Rossainz and Capel 2005-2):

An object manager (Figure.1) that it represents the CPAN in itself and makes of him an encapsulated abstraction that it hides their internal structure. The manager controls the references of a set of objects (a denominated object Collector and several denominated objects Stage) that represent the components of the CPAN and whose execution is carried out in parallel and it should be coordinated by the own manager.

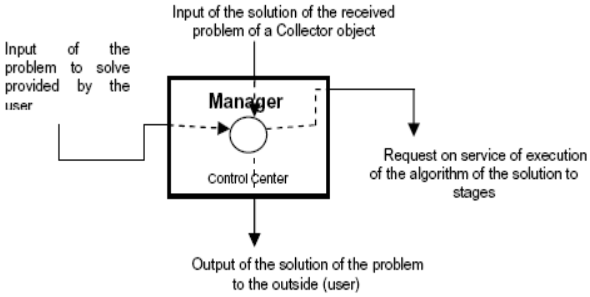

Input of the solution of the received problem of a Collector object
Input of the problem to solve provided by the user
Manager
Control Center
Request on service of execution of the algorithm of the solution to stages
Output of the solution of the problem to the outside (user)

Figure 1: Component MANAGER of model CPAN

The objects Stage (Figure.2) that are objects of specific purpose, in charge of encapsulating an interface type client-server that settles down between the manager and the objects slaves (objects that are not actively participative in the composition of the CPAN, but rather they are considered external entities that contain the sequential algorithm that constitutes the solution of a given problem), as well as providing the necessary connection among them to implement the communication pattern's semantics that seeks to be defined. In other words, each stage should act in parallel as a node of the graph that represents to the pattern. A stage can be directly connected to the manager y/o to

---

[2] For more details on these characteristics, to see section 2 of the present document

other component stage depending on the pattern peculiar of the implemented CPAN.

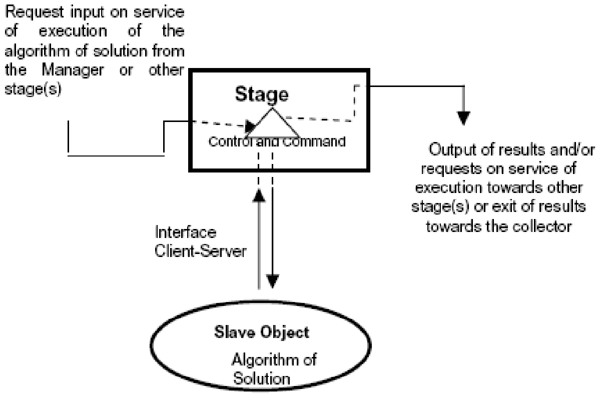

Request input on service of execution of the algorithm of solution from the Manager or other stage(s)
Stage
Control and Command
Output of results and/or requests on service of execution towards other stage(s) or exit of results towards the collector
Interface Client-Server
Slave Object
Algorithm of Solution

Figure 2: Component Stage of model CPAN and its associated slave object.

And an object Collector (Figure.3) that it is an object in charge of storing in parallel the results that he receives of the objects stage that has connected. That is to say, during the service of a petition, the control flow within the stages of a CPAN depends on the implemented communication pattern. When the composition concludes its execution, the result doesn't return to the manager directly, but rather an instance of the class Collector takes charge of storing this results and of sending them to the manager, which will send to the exterior the results, (that, in turn, send him an object collector), as soon as they go him arriving, without necessity of to wait to that all the results have been obtained.

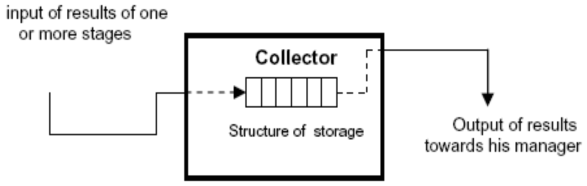

input of results of one or more stages
Collector
Structure of storage
Output of results towards his manager

Figure 3: Component Collector of model CPAN

5.3.1. Composition of the CPAN.

If we observe the scheme as a black box, the graphic diagram of the representation of a CPAN would be the one that is shown in Figure.4.

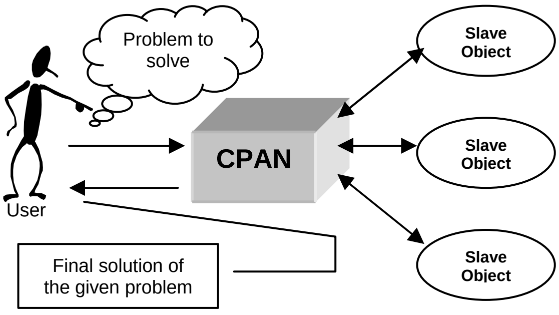


Figure 4: Graphical representation of a CPAN as black box

In summary, a CPAN is composed of an object manager that it represents the CPAN in itself, some objects stage and an object of the class Collector, for each petition that should be treated within the CPAN. Also, for each stage, an object slave will be taken charge of the implementation of the necessary functionalities to solve the sequential version of the problem that you pretend to solve (Figure.5).

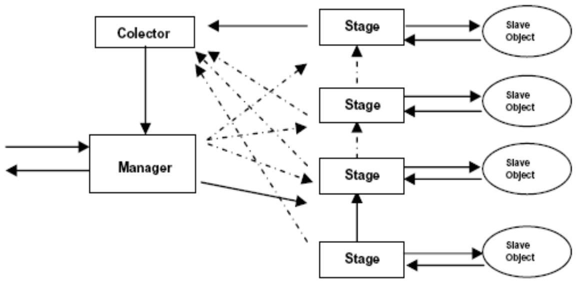


Figure 5: Internal structure of a CPAN. Composition of its components

The Figure.5 shows the pattern CPAN in general, without defining any explicit parallel communication pattern. The box that includes to the components, represents the encapsulated CPAN, the internal boxes represent compound objects (collector, manager and objects stages), as long as the circles are the objects slaves associated to the stages. The continuous lines within the CPAN suppose that at least a connection should exist between the manager and some of the component stage. The same thing happens between the stages and the collector. The dotted lines mean that it can have more than a connection among the components.

5.3.2. The CPAN seen as composition of parallel objects.

The objects manager, collector and stages are included within the definition of Parallel Object (PO) (Corradi 1995).

The Parallel Objects are active objects, that is to say, objects that have execution capacity in themselves. The applications within the pattern PO can exploit the parallelism so much among objects (inter-object) as the internal parallelism of them (intra-object). An object PO has a similar structure to that of an object in Smalltalk[3], but it also includes a politics of scheduling, determined a priori that specifies the form of synchronizing an or more operations of a class in parallel. The synchronization policies are expressed in terms of restrictions; for example, the mutual exclusion in processes readers/writers or the maximum parallelism in processes writers. All the parallel objects derive then of the classic definition of “class” more the incorporation of the synchronization restrictions (mutual exclusion and maximum parallelism). The objects of oneself class shares the same specification contained in the class of

---

[3] An Object in Smalltalk as in C++ is constituted up of a state and a behavior.

which you/they are instantiates. The inheritance allows deriving a new specification of one that already exists. The parallel objects support multiple inheritances.

5.3.2.1 Types of communication in the parallel objects.

The parallel objects define 3 communication ways: the synchronous communication way, the asynchronous communication way and the asynchronous future communication way.

1. **The synchronous way** stops the client's activity until the object active server gives him the answer. The notation[4]: `ref_obj.name_meth([lista_param])` it facilitates their use in the programming of applications.
2. **The asynchronous way** doesn't force the wait in the client's activity; the client simply sends the petition to the object active server and her execution continues. Its use in the programming of applications is also easy, because it is only necessary to create a thread and to throw it for its execution[5]. We will use the following notation to refer to this communication way: `Thread ref_obj.name_meth([lista_param]);` where `Thread` is thrown to execute the method `name_meth([lista_param])` of an object `ref_obj`.
3. **The asynchronous future way** makes only wait the client's activity when, within its code, the result of the method is needed to evaluate an expression. Their use is also simple, although its implementation requires of a special care to get a constructo with the wanted semantics. The notation used for it will be it: `FutureType futureVar = ref_Obj.name([lista_param])` that expresses the generation and future assignment of the result of a function invoked through a reference to an object. Where `FutureType` is the type that defines the future and `Anytype ResulVar = ref_Obj.futureVar`; it is used for the conversion of type of the future that returns the function when it is executed to a type `AnyType`. The word ANYTYPE is used to suggest the use of “any type”, the one that is of interest for the user.

The asynchronous and asynchronous future communication ways carry out the parallelism inter-objects executing the objects client and server at the same time.

5.3.3. Definition of the classes bases of a CPAN anyone.

As it has already been described, a CPAN comes from the composition of a set of objects of three types. In particular, each CPAN this compound for an object manager, some objects stage and an object collector for each petition carried out by the objects clients of the CPAN. Also, for each stage of the CPAN, an object slave will be taken charge of the implementation of the sequential part of the computation that is sought to carry out in the application or in the distributed and parallel algorithm. In PO the necessary basic classes to define objects manager, collector, stages and to compose a CPAN are:

- the abstract **class ComponentManager**
- the abstract **class ComponentStage**
- the concrete **class ComponentCollector**

An instance PO of a concrete class derived of the class ComponentManager represents a CPAN within an application (called manager) programmed according to the pattern of parallel object. The instances (called stages) of a concrete class derived of the class ComponentStage is connected to each other, to implement the stages composition. Each stage commands the execution of an object PO, called slave (slave) that is controlled by the own stage.

The creation of the stages and of the collectors and their later interactions are managed transparently to the code of the application by the manager. From the point of view of an user already interested in reuse the parallel behavior defined in some classes CPAN, the class of interest will be that of the manager. When an user is interested in using a CPAN within an application, he has to create an instance of a class manager specific, it is, one that implements the parallel behavior needed by the application and that it initializes it with the reference to the objects slaves that will be controlled by each stage and the name of the requested method. The following syntactic definitions have been written using the grammar free of context that is in the appendix A of the present document.

5.3.4. The Synchronization restrictions MaxPar, Mutex y Sync:

It is necessary having synchronization mechanisms, when parallel petitions of service take place in a CPAN, so that the objects that conform it can negotiate several execution flows concurrently and, at the same time, guarantee the consistency in the data that are processing. Within any CPAN the restrictions MAXPAR (The maximum parallelism or MaxPar is the maximum number of processes that you/they can be executed at the same time), MUTEX (The restriction of synchronization mutex carries out a mutual exclusion among processes that want to consent to a shared object. The mutex preserves critical sections of code and obtains exclusive access to the resources) and SYNC (The restriction SYNC is not more than a synchronization of the type producer/consumer of utility) can be used for the correct programming of their methods.

---

[4] The notations used in this section are based on the grammar of Parallel Objects described in appendix A.

[5] The POSIX Thread provides the instruction *pthread_create(. . .)* along with the type *pthread_t* for the creation and use of threads.

6. DESIGN AND CONSTRUCTION OF THE CPANS FARM, PIPE Y TREEDV

With the basic set of classes of the model of programming of PO they are possible to be constructed concrete CPANs. To build a CPAN, first it should be had clear the parallel behavior that one needs to implement, so that the CPAN in itself is this pattern. Several parallel patterns of interaction exist as are the farms, the pipes, the trees, the cubes, the meshes, the matrix of processes, etc.

Once identified the parallel behavior, the second step consists on elaborating a graphic of its representation as mere technique of informal design of what will be later on the parallel processing of the objective system; it is also good to illustrate its general characteristics, etc., and it will allow later to define its representation with CPANs, following the pattern proposed in the previous section.

When the model of a CPAN is already had concretized, that defines a specific parallel pattern; say for example, a tree, or some of those previously mentioned ones, the following step would be to carry out its syntactic definition and semantics.

Finally, the syntactic definition previous to a CPAN programmed is translated in the most appropriate programming environment for its parallel implementation. It would be verified that the resulting semantics is the correct one, it would be proven with several different examples to demonstrate its genericity and the performance of the applications would be observed that include it as a component software.

The parallel patterns worked in the present investigation have been the pipeline, the farm and the treeDV[6] to be a significant set of reusable patterns in multiple applications and algorithms. Being used at the moment with different purposes, in different areas and with different applications according to the literature that there is on the topic.

1. **The pipeline**, this compound for a set of interconnected states one after another. The information follows a flow from a state to another.
2. **The farm**, is composed of a set of worker processes and a controller. The workers are executed in parallel until reaching a common objective. The controller is the one in charge of distributing the work and of controlling the progress of the global calculation.
3. **In the treeDV**, the information flows from the root toward the leaves or vice versa. The nodes that are in the same level in the tree are executed in parallel making use of the denominated technique of design of algorithms it Divide and Conquer for the solution of the problem.

These parallel patterns conform the library of classes proposed within the pattern of the CPAN.

6.1 The Cpan PipeLine.

It is presented the technique of the parallel processing of the pipeline as a High Level Parallel Composition or CPAN, applicable to a wide range of problems that you/they are partially sequential in their nature. The CPAN Pipe guarantees the parallelization of sequential code using the patron PipeLine.

The Figure.6 represent the parallel pattern of communication Pipeline as a CPAN.

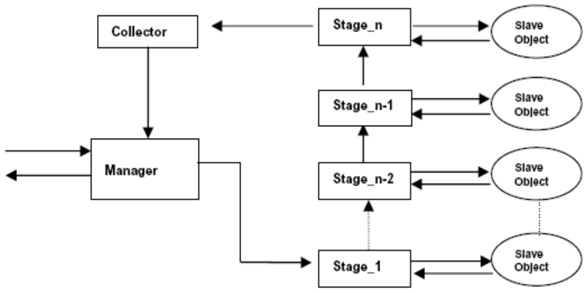


Figure 6: The CPAN of a Pipeline

The objects *stage_i* and *Manager* of the graphic pattern of the CpanPipe are instances of concrete classes that inherit the characteristics of the classes ComponentManager and ComponentStage.

6.2 The CPAN Farm.

It is shown the technique of the parallel processing of the FARM as a High Level Parallel Composition or CPAN.

The representation of parallel pattern FARM as a CPAN is show in Figure. 7.

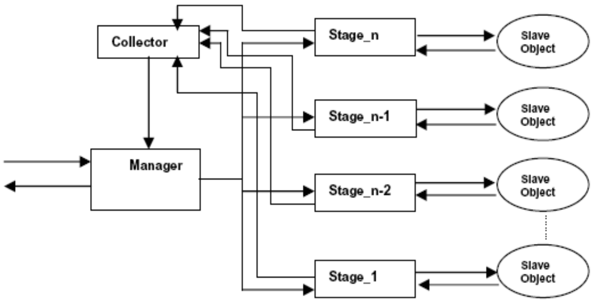


Figure 7: The Cpan of a Farm

The same as in the previous pattern, the objects Manager and *stage_i* are respectively instances of the classes that inherit of the classes base denominated ComponenManager and ComponentStage.

6.3 The Cpan TreeDV.

Finally, the programming technique is presented it Divide and Conquer as a CPAN, applicable to a wide range of problems that can be parallelizable within this scheme.

The representation of the patron tree that defines the technique of it Divide and Conquer as CPAN has their model represented in Figure. 8.

---

[6] The pattern treeDV implements the paradigm of programming of divide and conquer by means of the use of binary trees.

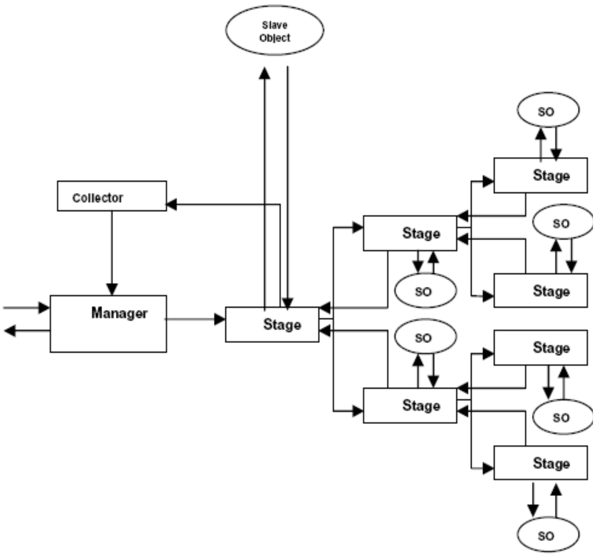


Figure 8: The Cpan of a TreeDV

Contrary to the previous models, where the objects slaves were predetermined outside of the pattern CPAN, in this model an object slave is only predefined statically and associated to the first stage of the tree. The following objects slaves will be created internally by the own stages in a dynamic way, because the levels of the tree depend from the problem to solve and a priori the number of nodes that can have the tree is not known, neither its level of depth.

These constitute a significant set of reusable communication patterns in multiple parallel applications and algorithms. See (Capel and Rossainz 2004; Rossainz 2005) for details.

7. USE OF A CPAN WITHIN AN APPLICATION

Once implemented the CPANs of interest, the way in that you/they are used in user's application is the following one:

1. It will be necessary to create an instance of the class manager of interest, that is to say, one that implements the required parallel behavior in agreement with the following steps:
   1.1. To initialize the instance with the reference to the objects slaves that will be controlled by each stage and the name of the method requested as an association of even (slave_obj, associated_method).
   1.2. The internal stages is created (using the operation *init()*) and they are passed each one an association (slave_obj, associated_method) that will use invoking the associated_method on their slave object.
2. The user asks the manager to begin a calculation through the execution within the CPAN of the method execution(). This execution is carried out as it continues:
   2.1. The object collector is created with respect to the petition.
   2.2. They are passed to the stages the input data (without verification of types) and the reference to the collector.
   2.3. The results are obtained from the object collector.
   2.4. The collector returns the results again to the exterior without verification of types.
3. An object manager has been created and initialized and some execution petitions can be dispatched in parallel.

8. RESULTS OBTAINED

Some CPANs adapt better to the communication structure of a given algorithm than others, therefore yielding different speedups of the whole parallel application. The way in which it must be used to build a complete parallel application is detailed below.

1. It is necessary to create an instance of the adequate class manager, that is to say, a specialized instance (this involves the use of inheritance and generic instantiation) implementing the required parallel behavior of the final manager object. This is performed by following the steps:
   1.1. Instance initialization from the class manager, including the information, given as associations of pairs (slave_obj, associated_method); the first element is a reference to the slave object being controlled by each stage and the second one is the name of its callable method.
   1.2. The internal stages are created (by using the operation *init()*) and, for each one, the association (slave_obj, associated_method) is passed to. The second element is needed to invoke the associated_method on the slave object.
2. The user asks the manager to start a calculation by invoking the *execution ()* method of a given CPAN. This execution is carried out as it follows:
   2.1. a collector object is created for satisfying this petition;
   2.2. input data are passed to the stages (without any verification of types) and a reference to the collector;
   2.3. results are obtained from the object collector;
   2.4. The collector returns the results to the exterior without type verification.
3. An object manager will have been created and initialized and some execution petitions can then start to be dispatched in parallel.

We carried out a Speedup analysis of the Farm, Pipe and TreeDV CPANs for several algorithms in an Origin 2000 Silicon Graphics Parallel System (with 64 processors) located at the European Center for Parallelism in Barcelona (Spain) this analysis is discussed below.

Assuming that we want to sort an array of data, some CPANs will adapt better to communication structure of a Quicksort algorithm than others. These different parallel implementations of the same sequential algorithm will therefore yield different speedups. The program is structured of six set of classes instantiated from the CPANs in the library High Level Parallel Compositions, which constitute the implementation of the parallel patterns named Farm, Pipe and TreeDV. The sets of classes are listed below:

1. *The set of the classes base*, necessary to build a given CPAN.
2. The set of the classes that define the abstract data types needed in the sorting.
3. *The set of classes that define the slave objects*, which will be generically instantiated before being used by the CPANs.
4. The set of classes that define the Cpan Farm.
5. The set of classes that define the Cpan Pipe.
6. The set of classes that define the Cpan TreeDV.

This analysis of speedup of the CPANs appears in Figures 9, 10 and 11. In all cases the implementation and test of the CPANs Farm, Pipe and TreeDV 50000 integer numbers were randomly generated to load each CPAN.

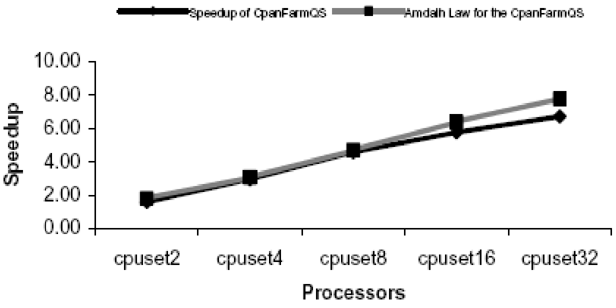


Figure 9: Scalability of the Speedup found for the CpanFarm in 2, 4, 8, 16 and 32 processors

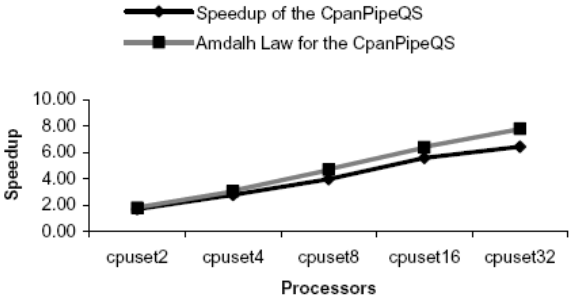


Figure 10: Scalability of the Speedup found for the CpanPipe in 2, 4, 8, 16 and 32 processors

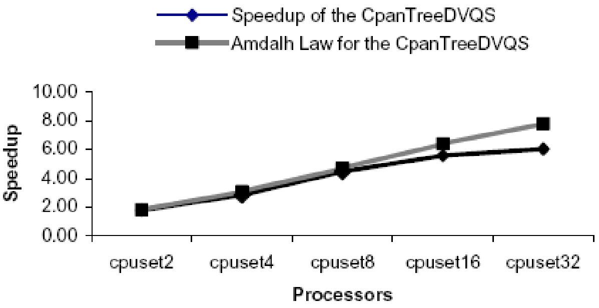


Figure 11: Scalability of the Speedup found for the CpanTreeDV in 2, 4, 8, 16 and 32 processors

9. CONCLUSIONS

1. A method of original programming has been developed based on High Level Parallel Compositions or CPANs
2. Patterns of communication/interaction have implemented themselves within the model of the CPAN commonly used in the parallel and distributed programming: the Cpan Pipe, the Cpan Farm and the Cpan TreeDV.
3. The implemented CPANs can be exploited, thanks to the adoption of the approach oriented to objects using the different mechanisms of reusability of the paradigm to define new patterns already using those built.
4. Well-known algorithms that solve sequential problems in algorithms parallelizable have transformed and with them the utility of the method has been proven and of the component software developed in the investigation.
5. The CPANs Pipe, Farm and TreeDV conform the beginning of the library of classes that intends in this work.
6. The restrictions of synchronization have been programmed of original form suggested by the model of the CPAN for their parallel and concurrent operation: the maximum parallelism (MaxPar), the mutual exclusion (Mutex) and the synchronization of communication of processes readers/writers (Sync).
7. Of equal it forms the programming in the asynchronous future communication way for results "futures" within the Cpans it has been carried out in an original way by means of classes.

REFERENCES

Bacci, Danelutto, Pelagatti, Vaneschi, 1999, SklE: A Heterogeneous Environment for HPC Applications. *Parallel Computing*, Volume 25, pp. 1827-52.

Brassard, G., Bratley, P., 1997, *Fundamentos de Algoritmia*, Spain, Prentice-Hall.

Brinch Hansen, 1993, Model Programs for Computational Science: A programming methodology for multicomputers. *Concurrency: Practice and Experience*. Volume 5, Number 5, 407-423.

Brinch Hansen, 1994, SuperPascal- a publication language for parallel scientific computing.

Concurrency: Practice and Experience, Volume 6, Number 5, 461-483.

Capel, M., Troya, J., M., 1994, An Object-Based Tool and Methodological Approach for Distributed Programming. *Software Concepts and Tools*, Volume 15, pp. 177-195.

Corradi, A., Leonardi, L., 1991, PO Constraints as tools to synchronize active objects. *Journal Object Oriented Programming*, Volume 10, pp. 42-53.

Corradi, A., Leonardo, L., Zambonelli, F., 1995, *Experiences toward an Object-Oriented Approach to Structured Parallel Programming*. Technical report no. DEIS-LIA-95-007. Italy.

Danelutto, M., Orlando, S., et al. *Parallel Programming Models Based on Restricted Computation Structure Approach*. Technical Report. Universitá de Pisa.

Darlington et al, 1993, Parallel Programming Using Skeleton Functions. *PARLE'93*, Munich.

Hartley, Stephen J., 1998, *Concurrent Programming. The JAVA Programming Language*. New York, Oxford University Press.

Lavander, Greg R., Kafura, Dennis G., *A Polymorphic Future and First-class Function Type for Concurrent Object-Oriented Programming*. Journal of Object-Oriented Systems. Available from: http://www.cs.utewxas.edu-users/lavender/papers/futures.pdf.

Roosta, Séller, 1999, Parallel Processing and Parallel Algorithms. Theory and Computation. *Springer*.

Rossainz, M., 1999, *Una Metodología de Programación Paralela en Java*. Technical Report. Universidad de Granada.

Capel, M., Rossainz, M., 2004. A parallel programming methodology based on high level parallel compositions. *14th International Conference on Electronics, Communications and Computers IEEE CS press*. México.

Rossainz, M., 2005, *Una Metodología de Programación Basada en Composiciones Paralelas de Alto Nivel (CPANs)*, PhD dissertation, Universidad de Granada.

Rossainz, M., Capel, M., 2005-2, An Approach to Structured Parallel Programming Based on a Composition of Parallel Objects. *XVI Jornadas de Paralelismo*, Granada, Spain, Thomson.

AUTHORS BIOGRAPHY

MARIO ROSSAINZ LOPEZ was born in Puebla, México and went to the University of Puebla, where he studied Sciences of Computation and obtained his degree in 1994. He works in the Faculty of Sciences of Computation of the University of Puebla from the year of 1995 where he is now in the research group of Software Engineering in the field of distributed systems. His e-mail address is: `mariorl@siu.buap.mx` and his Web-page can be found at `http://www.cs.buap.mx/~mrossainz`.

MANUEL I. CAPEL TUÑON was born in Spain and went to the University of Granada, where he studied Physics and obtained the MSC degree in 1982. He worked in the University of Murcia before moving in 1989 to the University of Granada where he is now leading a research group in the field of Concurrent Systems. His e-mail address is: `mcapel@ugr.es` and his Web-page can be found at `http://lsi.ugr.es/~mcapel`.

ANALYSIS OF TIME WARP ON A 32,768 PROCESSOR IBM BLUE GENE/L SUPERCOMPUTER

Akintayo O. Holder$^{(a)}$ **and Christopher D. Carothers**$^{(b)}$

$^{(a)(b)}$ Department of Computer Science
Rensselaer Polytechnic Institute
Troy, NY 12180, U.S.A.

$^{(a)}$holdea@cs.rpi.edu, $^{(b)}$chrisc@cs.rpi.edu

ABSTRACT

The aim of our work is to investigate the performance and overall scalability of an optimistic discrete-event simulator on a Blue Gene/L supercomputer. We find that strong scaling out to 16,384 processors is possible. In terms of event-rate, we observed 853 million events per second on 16,384 processors for the PHOLD benchmark. This is 1.5 times faster than any previously reported PDES synchronization protocol for PHOLD executing on a Blue Gene/L supercomputer (e.g. conservative, optimistic or hybrid). Additionally, we observed 2.47 billion events per second for a PCS telephone network model when executed on 32,768 processors. To the best of our knowledge, this is the first multi-billion event rate achieved for any Time Warp model executing on a Blue Gene/L supercomputer.

Keywords: simulation, high performance computing, parallel discrete event simulation

1. INTRODUCTION

Time Warp (Jefferson 1985) is a synchronization protocol that allows a parallel discrete-event simulation to speculatively process event computations, but if the synchronization mechanism detects an event that has been processed out of time stamp order (e.g. event causality error) it will undo or *roll back* the offending event computations. The most common technique for realizing rollback is *state-saving*. Here, the original value of the state is saved prior to event execution. Upon rolling back, the state is restored by copying back the stored value.

In this paper we make two contributions; we demonstrate that is possible to construct an efficient Time Warp simulator which achieves linear scalability on the Blue Gene/L supercomputer out to 16,384 processors and continues to increase its event rate out to 32,768 processors. We also demonstrate that a synchronous Global Virtual Time (GVT) algorithm, which defines a lower bound on any unprocessed or partially processed event, can scale to 10's of thousands of processors.

Rensselaer's Optimistic Simulation System (ROSS) is the basis for this experimental performance study. ROSS was originally a Time Warp simulation engine that was optimized for shared memory systems and is based on Georgia Tech Time Warp (GTW) (Carothers, Perumalla and Fujimoto 1999). This version, which we now call *ROSS-SM*, has some key features that are important for efficient Time Warp execution. First, ROSS-SM efficiently manages memory consumption (Carothers, Perumalla and Fujimoto 1999) for both forward as well as rolled back event computations. In particular, pointers to events are used whenever possible rather than creating duplicate copies. This feature eliminates the need for searches when performing event cancellation (e.g. direct cancellation (Fujimoto 1989)). Using reverse computation also reduces the amount of state saved and has been shown to dramatically improve parallel performance (Carothers, Bauer and Pearce 2000). Finally, GVT is computed using Fujimoto's algorithm (Fujimoto and Hybinette 1997), an asynchronous algorithm that uses a shared global flag to signal GVT computation.

The Blue Gene/L is a completely different class of parallel system. Here, modest computing nodes are connected by a low latency, high bandwidth interconnects, including an independent collective network. Broadcast latency is comparable to point to point messaging, making global reduction efficient. The Blue Gene/L does not allow processors to directly access remote memory, but it includes an efficient message passing framework using MPI (MPI 1994). Consequently, our challenge is to migrate our efficient shared-memory implementation into a highly optimized message passing system that is capable of scaling to supercomputing class processor counts.

The following issues are addressed as part of our implementation of ROSS on the Blue Gene/L which we call *ROSS-MPI*: (i) the sharing of events between processes, (ii) the impact of remote communication on memory consumption, (iii) the role of global virtual time computation on fossil collection, (iv) identifying unique events and ensuring stable, deterministic execution.

The remainder of this study is organized as follows: a brief overview of the IBM Blue Gene/L supercomputer is provided in Section 2. We then discuss the details of the ROSS-MPI implementation in Section 3 and present our synchronous GVT algorithm in Section 3.3. The performance results are presented in Section 4. Finally, related work is discussed in Section 5 with closing remarks in Section 6.

2. BLUE GENE/L SUPERCOMPUTER

The Blue Gene/L is an ultra large-scale supercomputer system that is capable of having 131,072 processors. The Blue Gene philosophy is that more powerful processors is not the answer when it comes to winning the massively parallel scaling war (Adiga et al. 2002). Instead, the Blue Gene architecture balances the computing power of the processor against the data delivery speed of the network. This led designers to create smaller, lower power compute nodes (only 27.5 KW per 1024 nodes) consisting of two IBM 32-bit PowerPCs running at only 700 MHz with a peak memory per node of 1 GB. A rack of Blue Gene is composed into 1024 nodes consisting of 32 drawers of 32 nodes in each draw. Additionally, there are specialized I/O nodes that perform all file I/O. Nominally there is one I/O node for every 32 compute nodes.

Interconnecting both drawers of nodes and racks are five specialized primary networks. The first is the point-to-point network which allows data to be sent between nodes. This network is a 3-D torus consisting of 12 directional links with a bandwidth of 175 MB/s each in the $+-x$, $+-y$ and $+-z$ directions. The latency of a point-to-point message is a function of the distance traveled between nodes. The 32,768 processor Blue Gene/L used in this study consists of 16 racks with each rack being a 32x32x1 torus yielding a network of 32x32x16. The max distance is the sum of half the distance for each direction which is 40 (e.g., $16+16+8$) leading to a max delay of 4 μs (i.e., each hop has a max delay of 100 ns).

In addition to the point-to-point network, there is a global collective network that enables data collection, reduction and redistribution to all nodes (or a subset) with a latency of 5 μs. As we will see in Section 3.3, this collective network is critical to Time Warp's ability to efficiently compute Global Virtual Time (GVT) and re-claim memory. We observe here that the collective network is able to compute a global reduction operation across *all processors* almost as fast as the single longest 1-way delay of the point-to-point network. This suggest that any GVT algorithm using the point-to-point network will not scale as well as using the collective network.

Next, there is an independent barrier network that is able to complete a barrier of a full 64K node Blue Gene/L system in less than 1.5 μs. Finally, there is a separate control network used to transmit system health information as well as an Gigabit Ethernet network which provides connectivity between I/O nodes and an external parallel file system.

For this experimental study, the Blue Gene/L housed within the Rensselaer Computational Center for Nanotechnology Innovations (CCNI) is used. This is a 16 rack Blue Gene/L system with 8 racks having 512 MB of RAM per node and the other 8 racks configured with 1 GB of RAM per node. The IBM XLC C compiler was used for all the results in this paper. We were able to take full advantage of the compiler's peak optimization level as well as architecture specific settings. Our specific compiler options where: `-O5 -qarch=440d -qtune=440`.

3. ROSS-MPI IMPLEMENTATION

In the ROSS implementation of the Time Warp protocol, the processor element (PE) is an abstraction of the physical processor which we realize as an MPI task. They are independent processes that have exclusive memory access and communicate via message passing. Events that are destined for a logical process (LP) on another PE are sent as MPI messages to the correct task. Each PE owns a number of LPs and uses a master scheduler to process events in time stamp order for all LPs assigned to that PE. Under the Time Warp protocol, models are implemented as parallel applications where an LP would be a logical thread of execution. The event handler is executed by the event's destination LP, as the LPs are responsible for processing and scheduling events. The models are built from LPs and events, while the PEs and KPs are architectural features used by ROSS. This allows the model to define parallelism independently of the processor count. The model provides the addressing protocol for routing events among LPs and PEs. Next, because each physical processor has it own memory, remote events are duplicated at the source and destination PEs. This duplicaton allows us to implement a rollback that spans across physical processors and separate memory address spaces. Remote events, and how they are handled will be important to the performance of ROSS-MPI. Last, a synchronous GVT algorithm was implemented that exploits the Blue Gene/L's collective network to achieve scalable performance.

Kernel processes (KPs) (Carothers, Bauer and Pearce 2000) improve the efficiency of fossil collection by aggregating events processed for a group of LPs into a larger list. ROSS-MPI maps the LPs to the KPs and the KPs to the PEs, but it is the responsibility of the model to assign LPs their identifiers. When a PE needs to schedule an event

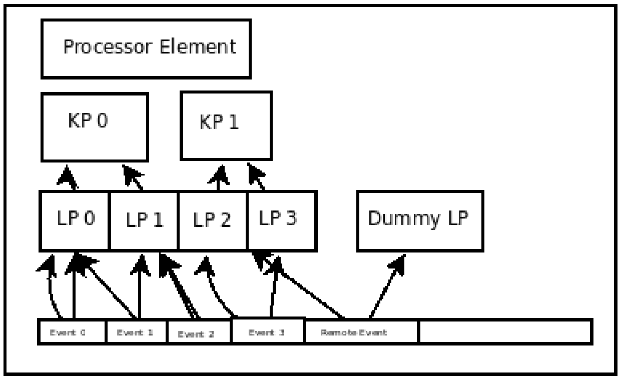


Figure 1: Architecture of a Processor Element (PE)

on another PE, it uses the model's LP addressing scheme, which will be described below. Remote events introduce the need to differentiate among events, so we use `age` a sequence number that uniquely identifies all the events generated between any pair of LPs. Because the MPI standard (MPI 1994) guarantees in order delivery of messages, event arrival order is preserved thus avoiding the complex case of having to process cancel events prior to the receipt of their positive event counterpart. This aspect is described in more detail in Section 3.2.1. Finally, the GVT algorithm empties the receive buffers and uses collective operations to ensure all processor elements agree that there are no transient events. We define an overflow buffer to be used for receiving remote events during GVT computation. This overflow buffer allows GVT computation to proceed when regular event memory has been exhausted.

3.1. Processor Element Communication

As shown in Figure 1, each MPI task contains a PE data structure, arrays of LPs and KPs, and a free list of events. The free list is an allocation of all the events that will be used over the life of the simulation. ROSS does not allocate memory during allocation, rather it provides a reference to an event in the free list. These are distributed equally among processor elements which should ensure that each processor performs an equal share of the work. ROSS-MPI assumes that the model will be well balanced, but better support for unbalanced models will be available in the future. This would include a more general LP to KP mapping structure, that will allow variance in the number of LPs assigned to a KP. We do believe that a balanced model is best suited for large platforms, like the Blue Gene. However, we acknowledge the need to support models where LPs are assigned varying workloads.

ROSS-MPI maps the LPs to KPs, but models may take advantage of the scheme when defining their LP addressing scheme. ROSS-MPI ensures that the i^{th} KP is assigned the $((i-1)(nlp/nkp)+1)$, $((i-1)(nlp/nkp)+2)..(i)(nlp/nkp)$ LPs, where nlp and nkp are the number of LPs and KPs respectively. Each LP is assigned an address that is comprised of an `index` and a `peid`, the model then assigns an `lpid`. The `peid` refers to the processor element that contains the LP, and the `index` is the location of the LP in the local array of LPs. The `lpid` is a unique identifier that is used by the model. The `map_lp_to_pe` function maps the `lpid` to the `peid` and the `map_lp_to_local` function maps it to the `index`. We chose to always compute the address pair consisting of the `index` and the `peid`, as opposed to constructing a table that describes the mapping or caching the computed results. This is a space-computation trade-off. A mapping table would limit the scale of the model that could be executed since mappings for all the LPs would be retained on each MPI task and caching would increase complexity of the implementation. Based on our current performance results, we have seen no indication that the use of efficient mapping functions degrades performance.

3.2. Remote Events

Remote events are generated when the source and destination LPs are not on the same PE. The source LP creates the remote event and places it on the current event's `caused_by_me` list. The "current event" is the event that is being processed by the LP during event generation. Next, the source LP sends the event to the PE that hosts the destination LP. When the destination PE receives an event it finds the correct destination LP, and inserts the event in the priority queue. If the source LP needs to rollback, it will send a remote cancel event that contains a duplicate of the remote event. ROSS only uses the source LP, destination LP, time stamp and age to identify the correct event, but a complete duplicate is sent. The mapping functions are used by the PEs to find the source and destination LPs of an event.

When a PE sends a remote event, it uses `map_lp_to_pe` to compute the destination `peid`. The destination PE, upon receiving a remote event, uses the `map_lp_to_index` to find the local LP that corresponds to the destination `lpid`. These mapping functions must initialized before the scheduler loop begins as any LP could be referenced once event processing commences. The model builder must ensure that all the `peid`, `index` pairs are mapped to `lpids`.

3.2.1. Augmenting Direct Cancellation

When a remote event or cancel arrives, the PE must be able to tell if the event is unique or a copy of an existing event. A remote cancel without the associated event is an error, as is the duplication of a remote event. With ROSS-SM every event is made unique by leveraging shared memory to perform direct cancellation (Fujimoto 1989). Here, a "cancel" message/event is a pointer reference to the actual real

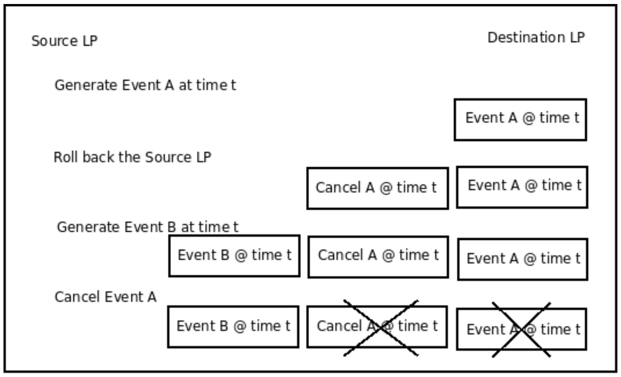


Figure 2: Handling concurrent events due to roll back

event, thus making the positive event and the anti-message the same object. With ROSS-MPI, the PE must search the priority queue and processed list to cancel a previously sent remote event. The use of KPs (Carothers, Bauer and Pearce 2000) allows us to perform a quick linear search of the processed list. The search of the priority queue, which is a Splay Tree data structure, is derived from the insert method which has a complexity of $O(log(n))$ (Sedgewick 1998). To implement a search we must be able to differentiate among the events in the simulation. Consider the case shown in Figure 2. Here, a source LP schedules an event A at time t to a destination LP. The source LP is then rolled back and it generates an anti-message for event A. After rollback is complete, the source LP schedules a new event B at time t on destination LP. When processing the cancel for A we must be aware that the new event B could have the same time stamp despite being the “correct” event. Thus, we need a mechanism to uniquely identify events that are to be cancelled. To accomplish this, we identify an event using these four fields: (i) the event’s simulated time at which it is to be processed `recv_ts`, (ii) source lp `src_lpid`, (iii) destination lp `dest_lpid`, and (iv) a sequence number `age`. Every logical process maintains an non-decreasing sequence counter, when it generates an event it assigns the event the current value as its `age`, and then increments the counter. This `age` will be different for each event generated by a given LP. In the scenario we discussed, event A and cancel A would have the same age while B, the newer event, would be different. This happens because cancels are copies of the original event, and only new events are given unique `ages`.

3.2.2. Correctness and efficiency

A remote cancel event is a copy of the original event that is sent with a different MPI tag. Using a stub event, like DSIM (Chen and Szymanski 2007), while reducing network traffic, appears to complicate event handling and may not improve memory efficiency. At the source PE, we fossil collect the remote event when their parent is being collected. We exploit the observation that only the parent has a reference to the remote event. When fossil collecting, we check the `caused_by_me` list and fossil collect all remote events. This approach allows us to reclaim memory efficiently and eliminates the need to manage lists of remote events.

ROSS-MPI retrieves remote events by combining blocking receives with non-blocking probes. The non-blocking probe signals when a remote event is available, but it will not block if the buffers are empty. The combination gives us the semantics of non-blocking receives with simpler code and without the use of expensive MPI operations like message cancel. We use the `MPI_ANY_TAG` to poll both event types, rather than retrieve events and cancels separately. This solves the following race condition: if the event and cancel arrive when the PE is retrieving cancels, the cancel, which is processed first, will appear to be unaccompanied. By polling both ports and checking the event type after retrieval the problem is avoided. This solution is dependent on the semantics of MPI point-to-point communication which guarantees that ”messages are non overtaking” (MPI 1994). The race condition is avoided since an event/cancel pair is sent between the same two LPs, and a cancel can only be generated after the initial event.

3.3. GVT algorithm

A consistent cut (Mattern 1994) divides the events into past and future. Here, no events would be sent from the future into the past. If all the processors agree to the cut, the Global Virtual Time (GVT) is the time stamp of the earliest event in the present.

Algorithm 1 is a global reduction (Chen and Szymanski 2007) GVT algorithm, it is synchronous and it uses collective operations to create a consistent cut. When the processors reach the synchronization point they ensure that all transient messages are accounted for by performing a collective sum over the count of outstanding messages. Once all messages are accounted for, the cut is formed and then the processors perform a collective minimum over their local virtual time (LVT) which consists of the minimum of any event in the priority queue or otherwise awaiting processing. The latency of broadcast on the Blue Gene/L makes a synchronous global reduction GVT algorithm an efficient choice.

3.3.1. Correctness

A correct GVT algorithm must solve the transient message and simultaneous reporting problems (Fujimoto and Hybinette 1997). We present arguments that show our algorithm addresses both problems. A transient message is a message that is not visible to any processor because it is traversing the network. The simultaneous reporting problem exists where both the sender and receiver expect the other to ac-

Algorithm 1 GVT Computation algorithm

```
Require: global_message_counter, the difference between
   events sent and received since last GVT.
Require: local gvt estimate, earliest remote event since the
   last GVT.
Ensure: GVT is the minimum of all unprocessed events
   message_counter = 0
   repeat
      while incoming messages available do
         read(remote event)
         decrement global_message_counter
         if local gvt estimate > remote event time stamp
         then
            local gvt estimate = remote event time stamp
         end if
         enqueue remote event
      end while
      MPI_Allreduce(global_message_counter,
      message_counter, SUM)
   until message_counter == 0
   global_message_counter = 0
   LVT = min(earliest event in priority queue,
   earliest unprocessed cancel event)
   MPI_ReduceAll(LVT,GVT,MIN)
```

count for the event in their respective notions of local virtual time (LVT).

Transient message, by induction. Basis: Consider the base case at time 0, no remote events have been created, so the sum of the differences between the messages sent and received is zero. There are no transient messages.

Inductive step: For a given epoch, we increase the counter for each message and decrease for each reception. At the end of the epoch the counter is zero. If a transient message exists, it would be sent during or before the current epoch. If the transient message was sent during this epoch, the counter must be negative when we entered the epoch. If the transient message was sent before, the counter must be positive when we entered. Since the counter is always zero when an epoch ends neither is possible. □

Simultaneous reporting, by contradiction. Assume the simultaneous reporting problem exists. This implies that a PE must have received the GVT computation request after it has transmitted the earliest event in the system and the destination PE already responded the GVT request and sent its LVT before it has received the earliest event. In this case, each PE believes the other responsible for the inclusion of this earliest event in there LVT calculation. So GVT is being computed while the event is being transmitted. Our prior argument shows, there are no messages in-flight when GVT computation commences. □

3.4. Memory Management

Memory management in ROSS-MPI focuses on the impact of remote events. ROSS-SM is a memory efficient Time Warp implementation but remote events have the potential to double memory consumption, as previously noted in Section 3.2. The pathological cases will always cause problems, but we take the following steps to manage memory consumption. First, overflow buffers are allocated for use when receiving events during GVT computation and, at the source PE we fossil collect remote events with their parents.

The overflow buffer is a one time allocation specified by the model, and is only used when the PE's free list has been exhausted. Events allocated from the overflow buffer are added to the PE's free list when deallocated, they are not returned to the overflow buffer. The number of events allocated to the overflow buffer is defined by `tw_events_gvt_compute`. The overflow buffer is only used for retrieving remote events during GVT computation, and its length should be dependent on the communication pattern of the model. It should be proportional to number of iterations through the full ROSS-MPI scheduler loops which is set by `g_tw_gvt_interval`, and the size of the processing batches ,`g_tw_mblock`. If the PE's free list is empty, it will abort the scheduler loop and begin to compute GVT, and then start fossil collection.

3.5. Eliminating Global Data Structures and Non-Deterministic Event Processing

In addition to efficiency and correctness, we ensure the simulator scales well by eliminating global data structures and globally redundant operations. Instead of a global array of random number generators, we distributed each random number generator as part of the LP data structure. The stream is reversible (Carothers, Perumalla and Fujimoto 1999), allowing the reverse computation to "undo" any previous calls to the random number generator. The new fields in the LP data structure are the arrays that contain the seeds of the random stream and the variables used in sampling other distributions. The streams are seeded sequentially, so given the `lpid` we can calculate how many seeds to skip and initialize the stream with limited redundant work. A nice side effect of this data structure design change is that it co-locates the random number seeds with the LP state which enables better LP data locality for improved cache performance during event processing.

We also address the increased probability that two events may have the same destination LP and time stamp. As discussed in (Wieland 1997) simulations with running times that are large compared with the granularity at which events are scheduled increase the probability that two or more events will have the same destination LP and time stamp, if not the same source LP. To ensure deterministic event processing in the face of event simultaneity, the source LP, destination LP and the age of an event will be considered when sorting events in priority queue. This allows every event to be uniquely identified, and ensures the ordering is independent of the insertion order. We also rollback when the new event is equal to or older than the last processed event. This insures that all events with the same time stamp will be executed in deterministic order, but at the expense of increased rollbacks. However, because this behaviour occurs infrequently in real applications, we do not believe overall performance is degraded. By rolling back under these conditions, we must consider the possibility of livelock if there is a cycle of LPs scheduling events with zero delay. We do not believe this pathological case to be a reasonable situation in a model.

4. EXPERIMENTAL RESULTS

PHOLD is a synthetic benchmark, commonly used for testing the performance of Time Warp simulators (Chen and Szymanski 2005) and (Perumalla 2006). PHOLD has minimal event processing, minimal look ahead due to event scheduling being based on a random distribution and a random communication pattern. PHOLD can be configured by changing the event population and the ratio of remote events. PHOLD was configured to schedule 10 percent of events to remote LPs. The simulations had 1024x1024 or 1,048,576 LPs and an initial event population of either 10 or 16 events per LP. The 10 events per LP case and 10 percent remote ratio are comparable with the PHOLD configuration used by Permulla's performance study in (Perumalla 2007).

The second workload model is a PCS network that provides wireless communication services for cellular phone subscribers. Here, the service area of a PCS network is populated with a set of geographically distributed transmitters/receivers called *radio ports*. A set of radio channels are assigned to each radio port, and the users in the *coverage area* (or *cell* for the radio port) can send and receive phone calls by using these radio channels. When a user moves from one cell to another during a phone call a *hand-off* is said to occur. In this case the PCS network attempts to allocate a radio channel in the new cell to allow the phone call connection to continue. If all channels in the new cell are busy, then the phone call is forced to terminate. It is important to engineer the system so that the likelihood of force termination is very low (e.g., less than 1%). What is special about this application is that it is an instance of a class of applications that are *self-initiated* (Nicol 1991). Here, LPs typically schedule most of their events to themselves, which leads to fewer remote messages relative to locally scheduled events making this class of applications well suited for ultra large processor count systems like the Blue Gene/L. For a detailed explanation of our PCS model, we refer the reader to (Carothers, Fujimoto and Lin 1995).

The PCS model had a grid of 4096x4096 PCS cells which results in LPs with a constant configuration except for the simulated time, which was doubled for the 32,768 processor run.

4.1. PHOLD Performance

Because our access to the Blue Gene/L is limited, we where only able to perform one run at each processor count configuration. However, as well will demonstrate, the Blue Gene/L's performance has very little variation unlike typical multi-user clusters and thus we believe these single run performance numbers to be very close to the multirun averages. Additionally, the CCNI Blue Gene/L partitions are predefined with the following node counts: 512, 1024, 4096, 8192 and 16384. Thus, in order to maxmially utlize the processor counts in each available partition class, we made runs using processor counts of 1024 using a 512 nodes, 2048 using 1024 nodes, 8192 using 4096 nodes, 16,384 using 8192 nodes and 32,768 using 16,384 nodes.

Unlike typical large-scale clusters, the Blue Gene/L has no virtual memory and does not support a multi-program environment where OS-level daemons co-exist and compete for resources with user-level compute jobs. Thus, we found very little variance in performance across multiple runs. In order to demonstrate repeatable performance, we used multiple runs on smaller processor partitions. Figure 3 shows the event rate of PHOLD over 10 runs with 2048 processors. The standard deviation of the event rate was 0.05% of the mean, with PCS (not shown) it was 0.01% of the mean.

Figure 4 shows the aggregate event rate of PHOLD as a function of processor count. For the 16 events per LP case, we observed a rate of 43 million events per second on 1024 processors, which increased to a peak event rate of 798 million on 16,384 processors. Additionally, we observed an increase in event rate for the 10 events per LP case. The peak event rate was 853 million. This is about 1.5 times better than the peak rate described by Perumalla (Perumalla 2007), using any PDES synchronization scheme (e.g. conservative, optimistic or hybrid) on the same hardware. We believe the increase in performance for the 10 events per LP case is due to lower priority queue overheads as consequence of the smaller per processor event population.

For the 10 events per LP case, the per processor event rate was 45,000 on 2048 processors and it increased to 52,000 with 16,384 processors. The speedup of PHOLD appears to be super-linear despite the use of strong scaling. A lack of available work was a concern given that the 1,048,576 LPs are spread over 16,384 processors yielding only 64 LPs per processor. We attribute the super-linear performance to a decrease in priority queue overheads as the processor count increases. Recall, we use a Splay Tree data structure for the priority queue which as both a $O(log(n))$ complexity for

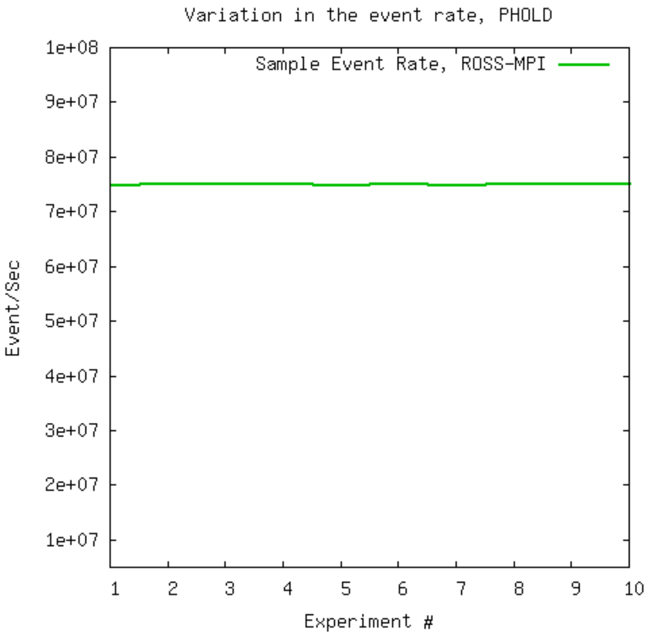


Figure 3: Event Rate for PHOLD across multiple runs on 2048 processors for 16 events per LP .

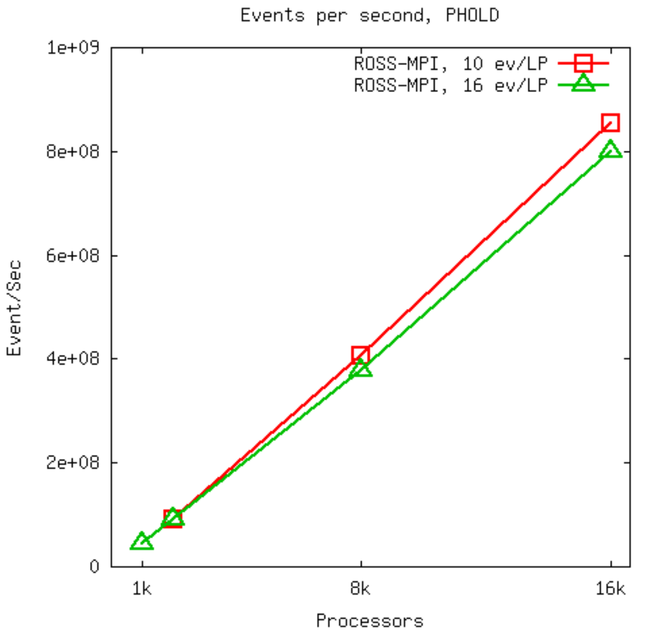


Figure 4: Event rate for PHOLD for both 10 events per LP and 16 events per LP as a function of processor count.

enqueue and dequeue operations. As the processor counts double, the event population per processor goes down by half, which decreases priority operations by a log(2) factor.

Figure 5 shows the number of rollbacks, primary and secondary for the 16 events per LP case as a percentage of committed events. This rollback ratio increased, almost linearly, with the increase in processor count. Both primary and secondary rollback ratios appeared to be gradually increasing, but we only observed a total rollback ratio of 0.54% with 16,384 processors. These low rollback ratios indicate the processors spend most of their time doing useful work.

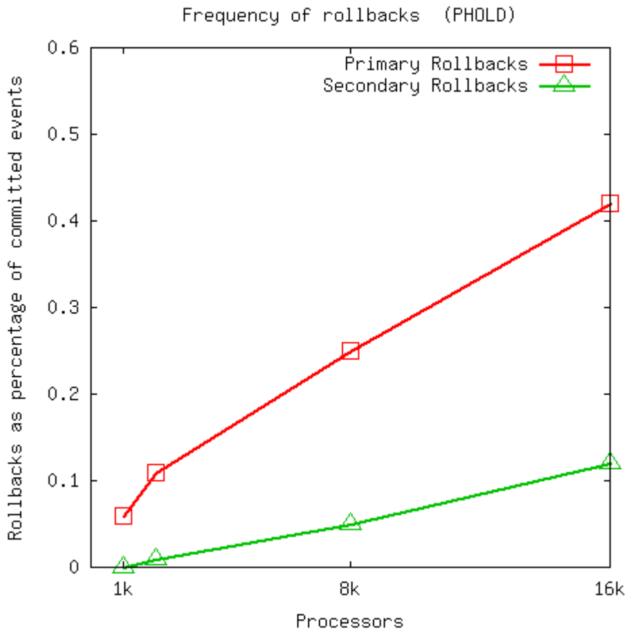


Figure 5: Primary and secondary rollbacks for PHOLD for 16 events per LP case as a function of processor count.

4.2. PCS Performance

Figure 6 shows that our event rate increases almost linearly to 2 billion events per second on 16,384 process, but beyond that the increase is sub-linear. The 25% increase in performance implies that the per processor event rate fell by 40% as we increased to 32,768 processors. The peak event processing rate was 2.47 billion events per second on 32,768 processors. Figure 7 shows that the ratio of rollbacks is less than 0.06% of the committed events. We do not believe changes in the ratios are significant as they remain low. The performance decrease could due to the size of the model, or limitations in the Blue Gene/L hardware, especially the collective network. The longer run for the 32K processor case was not believed to be responsible for the sharp decrease in per processor event rate. Further investigation is needed before we fully understand this phenomenon.

5. RELATED WORK

There have been some investigation into the performance of discrete event simulation on supercomputers with more than 1000 processors. The performance of conservative,

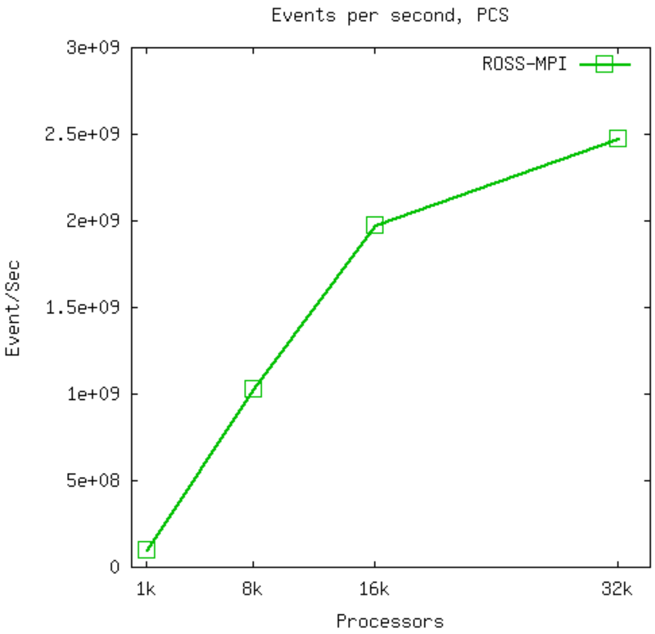


Figure 6: Event rate for PCS Model as a function of processor count.

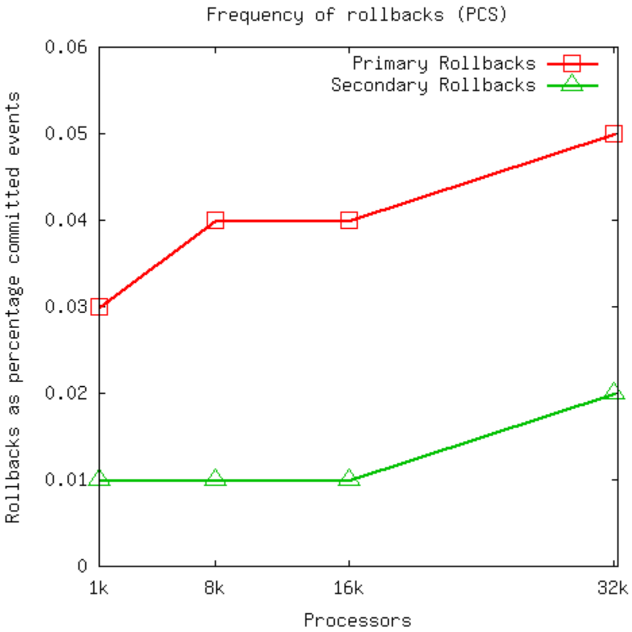


Figure 7: Primary and secondary rollbacks for PCS Model as a function of processor count.

optimistic and other approaches to PDES on the Blue Gene/L has been examined (Perumalla 2007). The observed performance may have been limited by the use of strong scaling and limited access to the Blue Gene/L. In a related Blue Gene Consortium report (Perumalla 2007), they describe porting the SCATTER road network model, highlighting the issue of porting existing models. PHOLD and PCS are well balanced models, we consider load balancing to be beyond the scope of this paper, but it would be interesting to consider problems where this is an issue.

The performance of PDES on a 750 node Alpha server has also been investigated (Chen and Szymanski 2005). They were able to process an impressive 228 million events per second on 1024 processors. Their PHOLD model schedules events at one of the four nearest neighbors, which should exploit the quad processor SMP nodes while avoiding the Quadrics network switch. DSIM uses, Time Quantum GVT (Chen and Szymanski 2007), a manager-worker GVT algorithm which reserves processors as GVT managers. They estimate that one manager is needed for every 128 processors, but the approach appears to be scalable.

In the context of GVT algorithms based on hardware-based acceleration approaches, there been some activity in the past. Of note, Pancerella and Reynolds (Pancerella and Reynolds 1993), they present results of simulations that suggest that hardware assisted, target-specific global reductions can dramatically improve parallel simulator performance. More recently Noronha and Abu-Ghazaleh (Noronha and Abu-Ghazaleh 2002) has shown the benefits of offloading the GVT computation to network interface cards. However, here Mattern's asynchronous GVT algorithm (Mattern 1994) is used as opposed to a synchronous reduction network-based algorithm.

6. CONCLUSIONS

We demonstrate that it is possible to construct an efficient optimistic Time Warp simulator which achieves linear scalability on the Blue Gene/L supercomputer, and we were able to process almost 2 billion events per second on a 16,384 processors for the PCS model. This would indicate that the optimistic approach to PDES has strong scaling potential on the largest scale supercomputers of today. It also demonstrates that a synchronous GVT computation algorithm can be used in an efficient Time Warp implementation provided that the underlying supercomputer architecture supports high-performance global collective operations. These results suggest that a synchronous algorithm, which can exploit the presence of a global collective/reduction network, will outperform a point-to-point solution, especially when scaling to near-petascale supercomputer system.

As future work, we plan to compare how the Blue Gene performs relative to other supercomputer architectures such as Texas Ranger, AMD Quad-Core Opteron Cluster and the Cray XT3 MMP system which are both part of the TeraGrid (see `www.teragrid.org`).

REFERENCES

Adiga, N.R and et al., 2002. An overview of the Blue Gene/L Supercomputer. *Proceedings of the ACM/IEEE Conference on Supercomputing*, pp. 1–22, November 16–22 Baltimore, Maryland, USA.

Carothers, C.D., Bauer, D. and Pearce, S., 2000. ROSS: A High-Performance, Low Memory, Modular Time Warp System, *Proceedings of the 14th Workshop on Parallel and Distributed Simulation*, pp. 53–60, May 28 - 31, Bologna, Italy.

Carothers, C.D., Fujimoto, R.M. and Lin, Y-B. 1995. A case study in simulating PCS networks using Time Warp. *Proceedings of the 9th workshop on Parallel and Distributed Simulation*, pp 87–94, June 13 - 16, Lake Placid, New York, USA.

Carothers, C. D., Perumalla, K.S., and Fujimoto, R.M., 1999. Efficient optimistic parallel simulations using reverse computation. *ACM Transactions on Modeling and Computer Simulation* 9(3):224–253.

Chen, G. and Szymanski, B.K., 2005. DSIM: scaling Time Warp to 1,033 processors. *Proceedings of the 37th conference on Winter simulation*, pp 346–355. December 4-7, Orlando, Florida, USA.

Chen, G. and Szymanski, B.K., 2007. Time quantum GVT: A scalable computation of the global virtual time in parallel discrete event simulations. *Scalable Computing: Practice and Experience:Scientific International Journal for Parallel and Distributed Computing*, 8(4):423–436.

Fujimoto, R.M. and Hybinette. M., 1997. Computing global virtual time in shared-memory multiprocessors. *ACM Transactions on Modeling and Computer Simulation*, 7(4):425–446.

Fujimoto, R.M., 1989. Time Warp on a shared memory multiprocessor. *Transactions of the Society for Computing Simulation International*, 6(3):211–239.

Jefferson, D.R., 1985. Virtual time. *ACM Transactions on Programming Language Systems*, 7(3):404–425, 1985.

Mattern, F., 1994. Efficient algorithms for distributed snapshots and global virtual time approximation. *Journal of Parallel and Distributed Computing*, 18(3):423–434.

Message Passing Interface Forum., 1994. *MPI: A message-passing interface standard.* Message Passing Interface Forum. Available from: http://www.mpi-forum.org/ [accessed 16 March 2008]

Noronha, R. and Abu-Ghazaleh, N.B., 2002. Using programmable nics for Time Warp optimization. *Proceedings of the International Parallel and Distributed Processing Symposium*, April 15-19, Fort Lauderdale, Florida, USA.

Nicol, D.M., 1991. Performance bounds on parallel self-initiating discrete-event simulations. *ACM Transactions on Modeling and Computer Simulation*, 1(1):24–50.

Pancerella, C. and Reynolds, P.F., 1993. Disseminating critical target-specific synchronization information in parallel discrete event simulation. *: Proceedings of the 7th Workshop on Parallel and Distributed Simulation*, pp 52–59, May 16 - 19, San Diego, California, USA.

Perumalla, K.S., 2006. *Ultra-scale parallel discrete event applications.* Oak Ridge National Laboratory. Available from: `http://www.bgconsortium.org/` [accessed 16 March 2008]

Perumalla, K.S., 2007. Scaling Time Warp-based discrete event execution to 10**4 processors on a Blue Gene supercomputer. *Proceedings of the 4th international conference on Computing frontiers*, pp 69–76, May 7-9, Ischia, Italy.

Sedgewick, R., 1998. *Algorithms in C.* 3rd ed. Boston:Addison-Wesley

Wieland, F., 1997. The Threshold of Event Simultaneity. *Proceedings of the Workshop on Parallel and Distributed Simulation*, pp 56–59, June 10 - 13, Lockenhaus, Austria.

AUTHORS BIOGRAPHY

AKINTAYO HOLDER is a Ph.D. student in the Computer Science Department at Rensselaer Polytechnic Institute. His research interests are parallel and distributed computing.

CHRISTOPHER D. CAROTHERS is an Associate Professor in the Computer Science Department at Rensselaer Polytechnic Institute. He received the Ph.D., M.S., and B.S. in Computer Science from Georgia Institute of Technology in 1997, 1996, and 1991, respectively. His research interests include parallel and distributed systems, simulation, networking, and computer architecture.

DISCRETE EVENT SIMULATION WITH UNIVERSAL PROGRAMMING LANGUAGES ON MULTICORE PROCESSORS

Thomas Wiedemann

University of Applied Science Dresden

wiedem@informatik.htw-dresden.de

ABSTRACT

The current hardware development is characterized by a in-creasing number of multi-core processors. The performance advantages of dual and quad core processors are already applied in high speed calculations of video streams and other multimedia tasks. This paper discusses possible applications of multi-core processors in discrete simulation. The implementation of parallel threads on more than one core requires massive changes in the software structure and software module interaction. Such changes are only possible inside the source code and can not be realized in COTS-simulation systems. The paper presents a special approach by using an assembler based, very fast multitasking routine combined with an additional multi-core runtime system. The basic system approach is realized with Standard C/C++ and Delphi-compilers and offers an high flexibility and a good runtime performance.

Keywords: Multicore processors, discret event simulation with universal languages

1. INTRODUCTION

The main algorithms and mathematical foundations of simulation systems are well defined and efficient (Wiedewitsch and Heusmann 1995). Nevertheless, the real application of simulation systems is still difficult (Kuljis and Paul 2000). Not more than 10% of all industrial firms use simulation tools by a number of reasons:

·The implementation of discrete event simulation mod-els with standard programming languages like C++ or Delphi is difficult. The main problem is the parallel execution of thousand or million small processes, which represent the simulation objects. The old concept of co-routine switching is not supported by modern programming languages.

·Especially in the area of optimization with simulation models exists a performance problem. It seems like a paradox, that an older simulation language like GPSS is significantly faster than modern simulation systems at run-time.

- In many areas of production planning it takes hours or days for finding useful solutions. Any speedup would improve the quality of simulation and optimization in real use cases.
- Commercial simulation packages like AutoMOD, Enterprise Dynamics, Arena or SLX are very complex. In fact of the small market for simulation, the prices of the systems are very high. Typical prices of more than $50,000 are too high for medium-sized firms.

In summary, it seems necessary to use cheap standard programming languages with fast scheduling algorithms on multi-core processor systems for much higher simulation speed and lower investment costs.

2. INSIDE DISCRETE EVENT SIMULATION

The foundations of discrete event simulation are already 30 years old. They are based on the basic principles of simulation, which are explained in detail in other papers (see former “How it works” sessions at the WSC, e.g. (Schriber 2003), (Kilgore 2001)).

In general, modeling and simulation of real world systems require parallel execution of a large number of processes in a specific order. This task is solved by all simulation systems. It is useful to discuss some details.

Process switching is the first task of parallel execution. The executing processor must switch from one process to an other process by preserving all states for future re-switching (see fig. 1). Often there are thousands of small processes with a high switching rate. Some operations are also conditionally. Switching inside basic functions is called co-routine switching. After a first realization in SIMULA such switching technologies were not integrated in C / C++ or similar languages. Other technologies, like pointer based functions calls and multi-threading are too slow and too complicated.

Process scheduling is the second task. The sequence of process switching must be determined by the simulation control unit. This is uncritical, if the schedule is simply determined by time or priority. It is critical, if the scheduling order depends on conditions, like blocking states in sequential organized queues.

Performance problems with simulation systems are often based on bad or non adequate switching and scheduling algorithms. Using standard multitasking algorithms from C/C++ or Delphi libraries are critical, because they are designed for switching a small number of large processes like tasks in operating systems. Often, the maximum number of threads is limited and the scheduling order can not be changed by the developer.

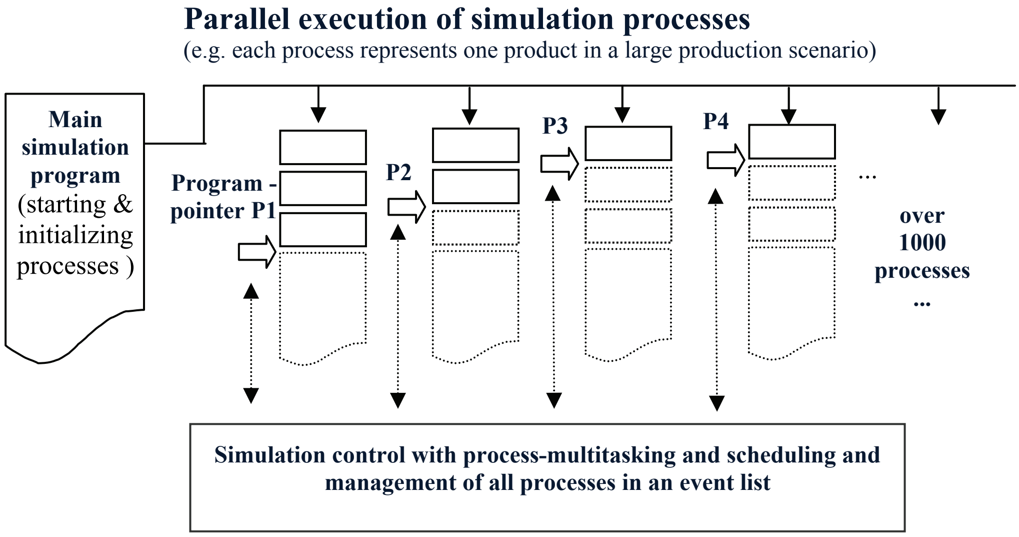

Parallel execution of simulation processes
(e.g. each process represents one product in a large production scenario)
Main simulation program (starting & initializing processes)
Program - pointer P1
P2
P3
P4
...
over 1000 processes ...
Simulation control with process-multitasking and scheduling and management of all processes in an event list

Figure 1 : Parallel execution and switching of simulation processes

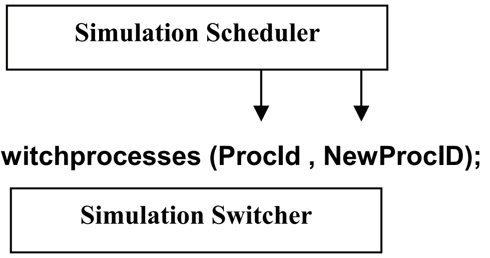
Algorithms for switching and scheduling define different requirements :

- process switching is a quite simple task and defines the main performance,
- process scheduling is quite complex, and less critical in performance.

Although it seems possible to develop a very efficient switching implementation, it is nearly impossible to develop a optimal scheduling algorithm for all applications, because there are dozens of scheduling algorithms on trees, sorted lists etc., which differ in terms of performance and complexity.

From this view, a main design decision was made: The **switching should be separated from scheduling** by using an open and flexible interface, which allows the simulation model builder a free choice of possible switching and scheduling modules.

Because of the fact, that nearly all existing computers are based on sequential (non-parallel) processors, the switching will always change from the current to the next process. If the scheduler has determined the next process, the switching will need only the information of the current and next process by using the following interface (see fig. 2).



```
switchprocesses (ProcId , NewProcID);
```

Figure 2 : Separation of switching and scheduling

This simple interface allows a wide spectrum of different switching and scheduling algorithms. The following pages will present some first implementations.

3. SWITCHING BY EXTREME MULTI-TASKING

3.1. Options for switching processes

The switching algorithm must save all local variables and the state of the processor of the current process, then he should load the new program and stack pointer address and must restore the processors register and local variables of the new process. Traditionally, the saving and restoring of the local variables is done by copying all memory blocks to backup areas, which is very time consuming.

Because of the fact, that in standard programming languages like C++ or Delphi all local variables are located on the stack, it seems possible to **switch all local data and the return address for the new process by only changing the current stack pointer address**. This simple change of the stack pointer value reduces the time for process switching significantly and allows very high rates of process multitasking. Otherwise there are some critical points of this approach:

- The change of the stack context is non trivial, because all local variables of all calling functions are switched off. In result, this method requires some special initialization of the stack during the start of each process.
- In general, the stack must provide memory space for an unknown number of functions calls. The size of stack space in standard implementations is between 16 Kbytes up to 64 Kbytes. The real used space is very different – efficient simulation functions need only some Hundred bytes of stack space, but Windows functions often require dozen Kilobytes of stack space. If any simulation process would use 64 Kilobytes of stack space, there would not be enough memory in the computer. For this reason the stack space is limited to 500 … 2000 Bytes per simulation process. If any simulation function calls an

expensive Windows function, this call is mapped to a larger stack space.

- Changing the stack pointer address could be dangerous for complex programming environments. The approach must be tested with each compiler and new version for avoiding stability problems.

In conclusion, the switching of processes by only changing the stack pointer is simple and very fast., but it has also some smaller disadvantages. For this reason, the attribute “extreme multitasking” is used to inform potential users about this specific approach.

3.2. Implementation results

The approach was tested by using DELPHI with the Object Pascal language. The stack pointer addresses are moved by assembler commands (see lines 7 – 10 of fig. 3) to and from a process address table. The push and pop commands save and restore the processor registers to the stack before switching. The number of POP/PUSH-operations depends on the specific processor and can change for other versions of compilers and languages.

```
procedure switchprocesses(OldProcId: integer;
NewProcID:integer );
begin asm  push eax // save calling environment
          push ebx
          push ecx
          push edi
          mov stackold,esp; end; // store old STACKP
      stacknew := cal[NewProcID];
      cal[OldProcId]:= stackold;
      asm mov  esp,stacknew; // get new STACKP
          pop edi
          pop ecx
          pop ebx
          pop eax  // get old environment
      end;
end; //AT THIS POINT THE SWITCHING HAPPENS !
```

Figure 3 : The code of the process switching module

Because of the fact, that there was no secure information about the possibility of changing the whole stack context by such a direct way, the author was impressed by the fact, that this code is also Debugger-safe. So if any application developer uses this code, he can still see all steps in step-wise execution: The old process enters this code sequence and after ending the switching code with the *end;* - statement (which is in practice a RETURN-assembler statement), the high level code–pointer will continue with the new process.

The necessary memory for this approach is simply the size of the stack of each process multiplied by the maximum number of processes. With a stack size of 2 Kilobytes about 500 processes are possible per Mbyte memory. If there are 100 Mbytes free memory, it allows 50.000 processes, which is a good value also for large models. If this size is too small, the simulation user should spend 100$ for an extra 1 Gigabyte RAM Memory.

In conclusion, we **PAY PERFORMANCE WITH MEMORY**, which is a cheap option today !

4. FLEXIBLE SCHEDULING

As defined by the interface (see fig. 2), the scheduler must select the next process for execution. This selection should be very fast for large numbers of processes and without long calculation times for inserting and deleting processes from the selection table. The kind of selection of course depends from the kind of simulation. In result, there will be different scheduling options for different simulation types.

Simple sequential scheduler

A simple sequential scheduler selects all processes one by one in the table and activates them. This kind of scheduler is only useful, if nearly all processes are executed in a strong periodical way. Related simulation models are used in traffic simulations, where all simulations objects (like cars or humans) are moving with small steps in every time step of simulation. The disadvantage of this scheduler is the bad performance in systems with very different activation rates.

Together with the switching module this scheduler allows a first test scenario for building up a simulation model. The resulting time for one whole cycle, measured over 1 Million switching / scheduling sequences was about 13 – 17 Nano-seconds on a 1,3 GHz Centrino PC and less than 10 Nano-seconds on a 2,5 GHz Desktop PC´s. In fact, that this time corresponds to about 30 basic assembler operations this cycle time seems to be the **lowest possible multitasking time cycle time**. Thread switching has cycle times from 500 ns up to some micro-seconds.

Future event list schedulers

For complex simulation models the sequential scheduler is not powerful enough. Better characteristics are possible with Future event list schedulers. They manage all processes in a sorted list. New processes are inserted by using their next activation time as the sort value. In result, the entry at the start of the list is always the next process for execution.

A simple list is critical for large amounts of processes, because the time for finding the place for insertion is linear growing with the number of processes. The current implementation task consists in finding algorithms with a better performance characteristic.

One option is an array-based tree with only 4 levels. In this scenario the time value is represented as a 32 bit long integer value. Each byte of this time is used as an index in one of the four levels (see fig. 5). With this approach, the insert time does not increase with a growing number of processes. The disadvantage is the same as before with the switcher – a high memory consumption. A test implementation shows, that about 3 Mbytes of RAM is necessary for running a typical production scenario.

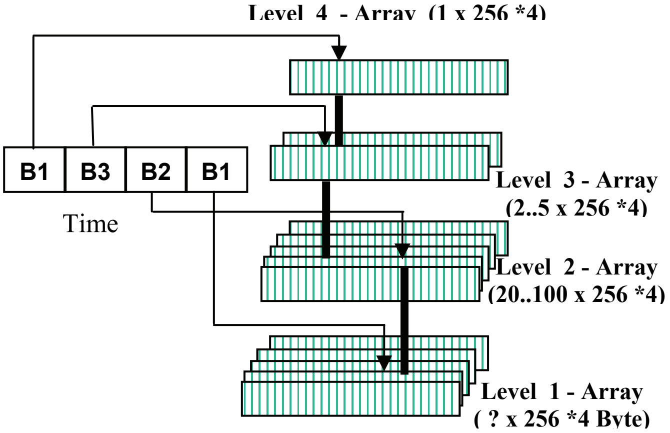

Level 4 - Array (1 x 256 *4)
B1
B3
B2
B1
Time
Level 3 - Array (2..5 x 256 *4)
Level 2 - Array (20..100 x 256 *4)
Level 1 - Array (? x 256 *4 Byte)

Figure 4 : An improved Future event scheduler

The main difference to existing simulation systems is the freedom of choice in the area of schedulers. While switching is assembler based and not very comfortable for High-level programmers, the development of new and much more improved scheduling algorithms is quite simple for experienced simulation kernel developers. After an initial time of building up different schedulers, the simulation user can select one of already existing schedulers. It is also possible to use different schedulers for different areas of a simulation model.

5. MULTI-CORE SUPPORT

The increasing number of multi-core processors in personal computers is very interesting also for simulation of large models, although it is not a new theme for the simulation community. Since many years distributed simulation is a well discussed topic in the simulation –community (see Perumalla 2006, PADS).
The main difference between the traditional distributed simulation and new opportunities of multi-core processors is defined by the wide availability of multi-core systems in the future :

- Instead of using specialized and very expensive hardware systems, nearly all future standard personal computers are equipped with 2,4 or more processor cores. So there is no cost overhead in hardware, when distributed simulation is used.
- Otherwise, standard computers are equipped only with standard operating system like Windows or Linux. In result, the implementation of distributed simulation must be realized with the methods of the existing operating system.

Implementation of distributed simulations on multi-core processors

The major number of multi-core systems will have two or four cores in the next few years. So the basic architecture of a distributed simulation should divide the algorithms on 2 or 4 or multiples of 2 cores.

The main experience from PADS-simulations shows, that a distributed execution of the simulation model itself is very complicated and the resulting speedup depends very heavily on the necessary communication between the distributed simulation modules. In bad cases, the speedup is below 1, which makes distributed simulation useless.

In the current situation with “only” 2 or 4 cores it seems more useful, **not to divide the model**, but to distribute the model and the simulation infrastructure. If there are more cores in the future, the cores should be used for a **pair based Hyper computing**, where one core is used for the simulation control and the other for the model. Of course also the traditional Hyper computing is possible and should be used if faster. Beside the model execution the simulation system must realize the following tasks:

- Scheduling of simulation processes with Future and Current event lists,
- Generation of a wide spectrum of random numbers (some random number types are quite expensive in terms of mathematical calculations)
- Storage of simulation results with basic statistical calculations (mean, standard deviation etc.) and compression of time series values.

On a 2-core system these tasks will be executed on the first core and the simulation model on the second core (see fig. 5). On a 4-core system the simulation control tasks will be executed on cores 1-3 and the simulation model on the fourth core (see fig. 6).

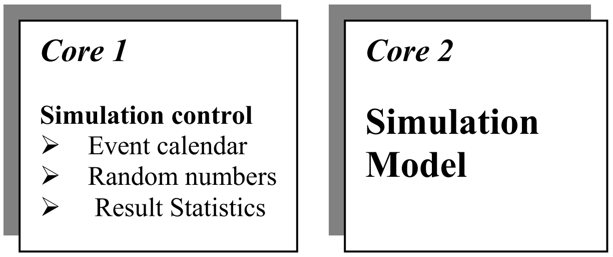


Figure 5 : 2-Core distributed simulation

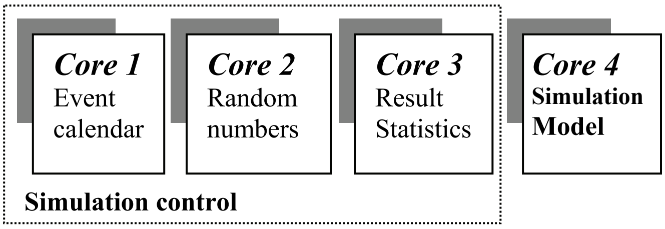


Figure 6 : 4-Core distributed simulation

The possible speedup of such an architecture depends very on the ratio between the model execution and the simulation control execution. In cases with small model functions, e.g. only random number based simulation of machining processes the time of model and simulation control execution could be nearly the same and the speedup could reach the number of existing cores 2 or 4, which means 50% or 75% less execution time.

In complex models with long running model functions the ratio between model and simulation control could be bad, so the speedup will decrease. In this case the cores should be used for a traditional Hyper computing, where each core executes one replication. In the case of Hyper computing the speedup equals nearly the number of processors. In applications where the speedup must be guaranteed, the usage of such parallel running simulations is the best and safe way. The described split of simulation control and model control seems only a interesting way of distributing simulation without dividing the models in very different and difficult ways.

Some additional measures could increase the speedup in a case with an oscillating ratio between model and simulation control:

- The generation of random numbers could be done in advance. So the next 200 or more random numbers could be generated and a model function with a burst usage (e.g. a Monte Carlo scenario inside a standard model) could use the numbers without waiting.
- The management of the event calendars is focused on delivering the next simulation events to the model. The storage of future events is of lower priority and is done after extracting the next future events from the list. A small secondary future event queue is possible.

If the simulation model runs always longer than the simulation control , some time expensive algorithms from the model (e.g. path-finding algorithms over a network or interpreting user-defined code) can be moved toward an free processor core.

In result of this options the ratio can be fine tuned towards similar time of model and simulation control execution , which maximizes the speedup. With some additional effort, this fine tuning can be automated in future systems.

First implementation results

The current simulation system is based on two (in the future also 4 or more) program threads. The first thread is started on the first processor and manages the simulation control functions. The second thread is started by the first thread and executes the simulation model. The interface between the threads is realized with shared memory.

The measurement of the speedup is quite simple : a first run is started with both threads only on one processor – which gives the single sequential time. The second run is executed with distributed threads on two cores and gives the time for distributed simulation. The first experiments with some simple queuing models with two lines of 4 machines show **speedups between 1.3 and 1.7** without special optimizations. Further work will analyze the effects of improvements of the interface and the discussed optimization measures.

6. THE SIMSOLUTION SYSTEM

All described basic routines will generate the kernel for a larger simulation environment, called “SIMSOLUTION”. The whole picture of the future “SIMSOLUTION”-simulation environment shown in Figure 6 and is based on former development of the author (Wiedemann 2000, Wiedemann 2002). Above the Code-level are the GUI-interfaces or interfaces to other information systems. Possible interfaces could be traditional desktop forms or web based forms in a internet browser. The large block in the center of the system controls all processes. It is also an interfacing layer between the specific tools at the tool level and the universal and standardized modules at the Model level. The communication between all modules is based on file or network techniques. The communication protocol uses XML-coded information. In many cases the content of the XML-databases or XML-encoded simulation results is only wrapped by an additional XML-layer and transported over the network. Larger amount of data, for example simulation results, will be compressed by well-known compression algorithms for better transportation speed. For the end user this data conversions will be transparent. Data and model storage is realized with data bases, where a universal canonical data model is used for all simulation model. By using SQL-statements the elements of the model could be manipulated also group wise. This option allows quick and efficient changes of large simulation models.

7. SUMMARY

The application of a universal programming language as basic language offers new opportunities for the development of discrete event simulators.

Especially new hardware options like multi-core processors could be used without long waiting for new versions of COTS-simulation systems. The applied architecture of a distribution between simulation model and simulation control is not every time the best option, but it guaranties in opposite to traditional distributed models in any case a speedup larger than one. But if possible and useful, also the model could be distributed on future processors with more than two cores.

Well-known programming languages like C, C++ or PASCAL will reduce the learning effort and offer better flexibility than traditional simulation systems. Adding new functions or interfacing to database system or new web-based technologies like Web-services is less expensive.

An additional effect are low investment cost also in multi core environments, because the run time modules are free of charge.

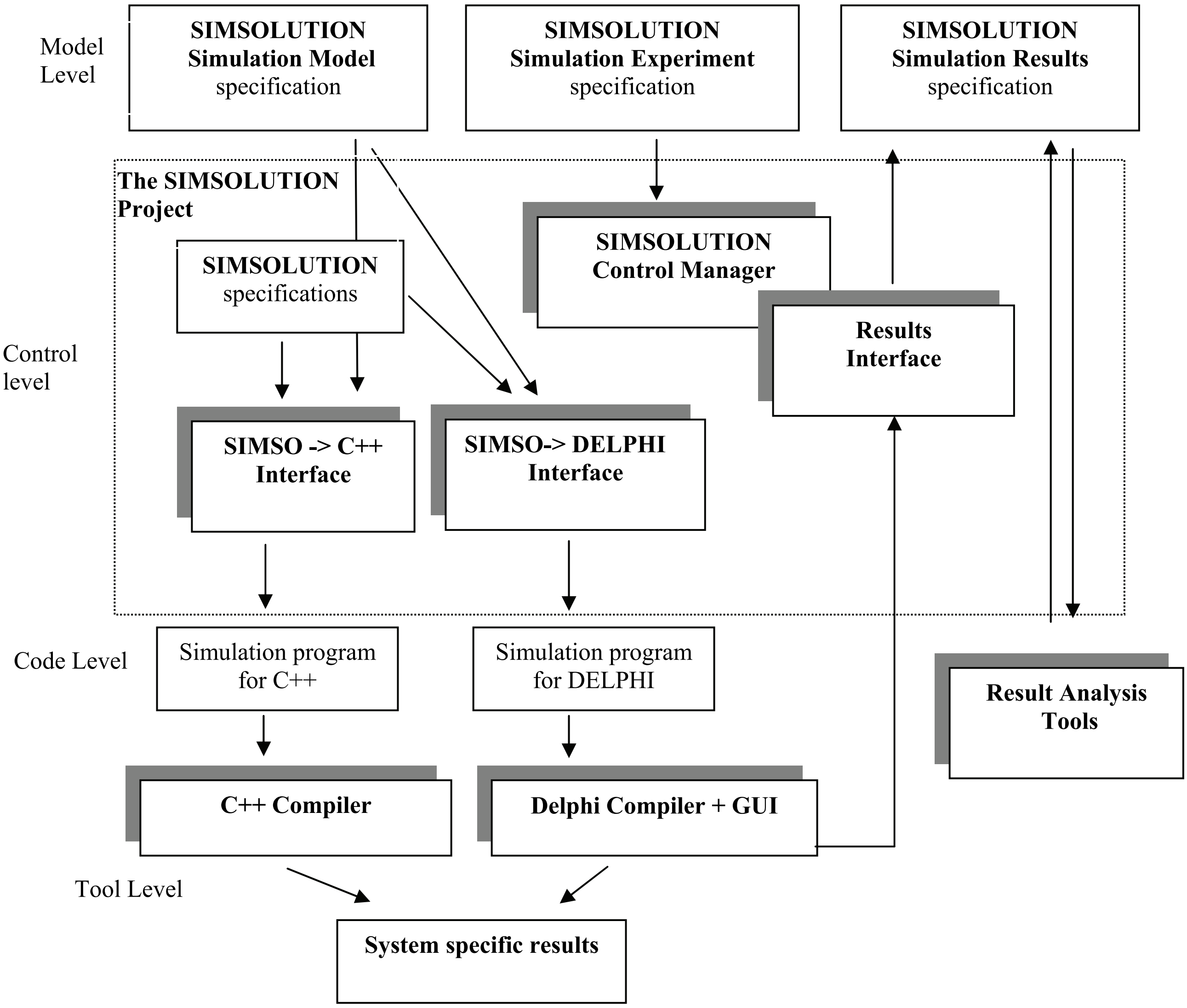

Model Level
SIMSOLUTION Simulation Model specification
SIMSOLUTION Simulation Experiment specification
SIMSOLUTION Simulation Results specification
The SIMSOLUTION Project
SIMSOLUTION specifications
SIMSOLUTION Control Manager
Results Interface
Control level
SIMSO -> C++ Interface
SIMSO-> DELPHI Interface
Code Level
Simulation program for C++
Simulation program for DELPHI
Result Analysis Tools
C++ Compiler
Delphi Compiler + GUI
Tool Level
System specific results

Figure 5 : The main architecture of the SIMSOLUTION - System

The usage of some specific Assembler-routines for switching could be seen as some disadvantage. But the resulting simulation speed is very high and offers new solutions especially in the area of optimization and simulation. For that reason, the current goal of development is to make the SIMSOLUTION-system some of the fastest simulation systems, even if there are some disadvantages or missing functions compared to other simulation systems.

In order to reduce the efforts for generating simulation models, the underlying programming language is managed by a universal modeling system, which generates universal, language independent XML-descriptions.

Code parsers and generators convert SIMSOLUTION-models to programs in C++, Delphi or .NET-languages.

In the future, with two sequential transformation processes a simulation model can be transferred between different platforms without manual changes.

Its future development will provide a universal and open simulation system. Any interested simulation expert or user is invited by the author for sharing his ideas, experience and cooperation inside the SIMSOLUTION-consortium.

REFERENCES

Kilgore, R. A. 2001. Open source simulation modeling language (SML*). In Proceedings of the 2001 Winter Simulation Conference*, ed., B. Peters, J. Smith. Piscataway, NJ: 2001

Kuljis, Jasna and Ray J. Paul, 2000: A Review of web based simulation: whiter we wander?, *Proceedings of the 2000 Winter Simulation Conference*, Orlando Florida, page 1872-1881

Jacobs, Peter, 2004: The DSO Simulation System. *Proceedings of the European Simulation Symposium*, Budapest, Hungary, October 2004

Perumalla K., 2006 "Parallel and Distributed Simulation: Traditional Techniques and Recent Advances". *Proceedings of Proceedings of the 2006 Winter Simulation Conference*

PADS : Website of the annual workshop for "Principles of Advanced and Distributed Simulation" http://www.pads-workshop.org/

Phillips, Lee Ann 2001. Special Edition using XML. Que Bestseller Edition, 2000

Schriber, Thomas J.; Brunner , Daniel T. : Inside Discrete-Event Simulation Software: How It Works and Why It Matters *Proceedings of the 2003 Winter Simulation Conference*, December 7-10, 2003, New Orleans, LA

Wiedemann, T., 2000. VisualSLX – an open user shell for high-performance modeling and simulation, *Proceedings of the 2000 Winter Simulation Conference*, Orlando Florida, pp. 1865-1871

Wiedemann, T., 2002. Next generation simulation environments founded on open source software and XML-based standard interfaces, *Proceedings of the 2002 Winter Simulation Conference*

Wiedewitsch J.; and Heusmann J. 1995. "Future Directions of Modeling and Simulation in the Department of Defense", *Proceedings of the SCSC'95*, Ottawa, Ontario, Canada, July 34-26, 1995

AUTHOR BIOGRAPHY

THOMAS WIEDEMANN is a professor at the Department of Computer Science at the University of Applied Science Dresden (HTWD). He has finished a study at the Technical University Sofia and a Ph.D. study at the Humboldt-University of Berlin. His research interests include simulation methodology, tools and environments in distributed simulation and manufacturing processes. His teaching areas include also intranet solutions and database applications. Email : <wiedem@informatik.htw-dresden.de>

TEMPLATES FOR DISTRIBUTED AGENT-BASED SIMULATIONS ON A QUASI-OPPORTUNISTIC GRID

Laszlo Gulyas[(a,b)], **Walter de Back**[(b)], **Gabor Szemes**[(b,a)],
Krzysztof Kurowski [(c)], **Werner Dubitzky** [(d)], **George Kampis** [(b)]

(a)AITIA International Inc, Budapest, Hungary
(b)Collegium Budapest, Hungary
(c)University of Queensland, Brisbane, Australia
(d)University of Ulster, Ireland

lgulyas@aitia.ai, wdeback@colbud.hu, gszemes@aitia.ai,
k.kurowski@imb.uq.edu.au, w.dubitzky@ulster.ac.uk, gkampis@colbud.hu

ABSTRACT

Complex systems are defined as systems with many interdependent parts which give rise to non-linear and emergent properties. Supercomputers constitute the *de facto* technology to deliver the required computational performance. However, supercomputers involve considerable costs, which many organizations cannot afford. The working assumption of this paper is that a *grid* could be enhanced by suitable middleware to provide features similar to those of supercomputers. However, simulation developers will face additional difficulties when adopting their applications to the grid. That is because the underlying topology of the computational infrastructure is dynamic. This paper reports on an ongoing effort to develop templates for distributed simulations on the grid with the integration of the Repast and ProActive packages.

Keywords: grid computing, agent-based simulation, opportunistic supercomputing, complex systems simulation templates

1. INTRODUCTION

Complex systems are defined as systems with many interdependent parts which give rise to non-linear and emergent properties determining the high-level functioning and behavior of such systems. Due to the interdependence of their constituent elements and other characteristics of complex systems, it is difficult to predict system behavior based on the 'sum of their parts' alone. Examples of complex systems include bee hives, bees themselves, human economies and societies, nervous systems, molecular interactions, cells and living things, ecosystems, as well as modern energy or telecommunication infrastructures. Arguably one of the most striking properties of complex systems is that conventional experimental and engineering approaches are inadequate to capture and predict the behavior of such systems.

To complement the conventional experimental and engineering approaches, computer-based simulations of complex natural phenomena and complex man-made artifacts are increasingly employed across a wide range of sectors. Agent-based simulation is a suitable and useful modeling paradigm for the decomposition and study of complex systems. (North and Macal, 2007) Typically, such simulations require computing environments which meet very high specifications in terms of processing units, primary and secondary storage, and communication. Supercomputers constitute the *de facto* technology to deliver the required specifications. Acquiring, operating and maintaining supercomputers involve considerable costs, which many organizations cannot afford. The working assumption of this paper (following that of the QosCosGrid project (Coti et al, 2008), http://www.qoscosgrid.eu/) is that a *grid* could be enhanced by suitable middleware to provide features and performance characteristics that resemble those of a supercomputer. We refer to such a grid as *quasi-opportunistic supercomputer*. The QosCosGrid project aims at developing such a system.

Computational simulations on supercomputers or on grid systems naturally require a *distributed* implementation. However, this complicates the model development significantly, especially, in cases of experimental, incremental model development, where the model structure may change dramatically during development. Moreover, the implementation complexity may go beyond the capabilities or interests of researchers in complex systems.

To face these issues, we are developing integration between the agent simulation toolkit Repast (North et al, 2006) and ProActive (Baude et al, 2006), a middleware for multi-threading, parallel and distributed computing, currently being interfaced with the QosCosGrid. This combination facilitates the design and deployment of distributed agent-based simulations over multiple computational nodes on multi-core machines, local clusters, and grid environments.

Naturally, the distribution and parallelization of computer programs, and thus simulations cannot be fully automatized. Parallelization is a multi-dimensional optimization problem. Two of the major dimensions are the minimization of communication between nodes, and the balancing of the processing load among the nodes. Except for the simplest programs, these are non-trivial issues that require in-depth knowledge about the workings of the particular program (simulation). However, general schemas for solutions exist, especially if the priority among the two major dimensions above is defined.

Our approach is to focus on the *minimization of communication*. We have identified classes of simulations that share common (agent-to-agent) communication templates. We are currently in the process of developing distributed and parallel Repast/ProActive implementations for these templates that can be subclassed and customized for particular user simulations. (Load balancing and other issues are handled, optionally, within the framework of the selected template.) The supported templates range from embarrassingly parallel applications such as parameter sweeps, to cellular automata, to static and dynamic (communication) networks, and to agents moving in abstract spaces.

The paper overviews the project's main directions and presents the current status, including working examples and prototypes.

2. THE QOSCOSGRID

The term *supercomputer* typically refers to a dedicated special-purpose multiprocessor computing system that provides close to best achievable performance for demanding parallel workloads. Supercomputers have several characteristics that enable them to efficiently execute considerable computational loads.

1. High-end and highly reliably hardware components, such as processing units, primary and secondary memory, and interconnects
2. Supercomputer middleware provides a straightforward abstraction of a homogeneous computational and networking environment, automatically allocating resources according to the underlying networking topology
3. The resources of a supercomputer are managed exclusively by a single centralized system, which enforces global resource utilization policies, thus maximizing hardware utilization while minimizing the turnaround time of individual applications.

These characteristics endow supercomputers with unprecedented performance, stability, and dependability properties.

Grid computing systems could be viewed as large-scale computing systems with considerable levels of hardware resources but with a lack of the features that make supercomputers so powerful. In particular, grids usually lack sophisticated support for highly parallel applications with significant inter-process communication requirements. Grid computing environments are based on heterogeneous, widely dispersed and time-variant resources which typically lack central control. Connected via local and wide area networks, grids typically rely on an opportunistic marshaling of resources into coordinated action to meet the needs of large-scale computing applications. Grids are often offered as panacea for all kinds of computing applications, including those that require supercomputing-like computing environments. However, this vision of grids as virtual supercomputers is unattainable without overcoming the performance and reliability issues plaguing current grids.

The main challenge of the QosCosGrid project is to overcome the current limitations of grids and implement a virtual computer which could be considered a viable approximation of a real supercomputer. The details of the proposed technical solutions are reported elsewhere. (Coti et al, 2008)

2.1. Requirements of Complex Systems Simulations

Despite the variety of scientific, engineering and other areas in which complex systems need to be studied and modeled, the key information technology requirements for computational modeling and simulation of complex systems are essentially identical across many domains and applications. These requirements include:

- Integration of large heterogeneous volumes of data and information that may arise from simulation or other information systems, which may be geographically widely dispersed.
- Design and execution of compute- and memory-intensive simulation programs may require resources that are not available locally.
- Handling the considerable volumes of data generated as output from the underlying simulations. These data need to be managed, analyzed and then shared using varied computational methodologies.

To address these requirements and achieve the main aims of the QosCosGrid project, the project is structured into the following key objectives:

- First, to provide a quasi-opportunistic supercomputing grid architecture and infrastructure which includes (i) Necessary grid middleware services including monitoring and measurement capabilities, (ii) User interfaces that enable easy access and use of resources by hiding the underlying complexity of the system, (iii) Flexible fault-tolerant message passing libraries, (iv) Data distribution enabling technology, and (v) Remote steering capabilities;
- Second, to develop services that provide (i) Dynamic resource brokering giving the best quality-of-service to any given complex system simulation, (ii) Reservation and orchestration of resources, communication, synchronization and routing as known from massively parallel processors computers.
- Third, to validate the quasi-opportunistic supercomputing concept with various types of complex systems simulation applications including (i) Research into the non-trivial parallelization of the simulation- and data-

processing applications typically encountered in the CS research, and (ii) to adapt the underlying algorithms to the quasi-opportunistic supercomputing environment.

This paper is focused on the last two items of the above ambitious set of goals.

3. COMPLEX SYSTEMS SIMULATIONS ON THE GRID

In this section, we provide a general abstract description of complex systems simulations that captures essential properties influencing the distributed implementation of such systems. As with every abstraction, certain details of individual cases are omitted, but to our belief, without the loss of generality.

(1) Interaction topology. A complex system consists of a finite set of interacting components. In our abstract treatment, each component will have a single state variable and an update function. (Notice that the single state variable can, in practice, be a combination of any finite number of variables.) The update function depends on the state variable itself and on the states of a subset of the other components. (The update function can be deterministic or probabilistic, in which latter case it yields a probability distribution of the next state.) The update functions' dependence on other components defines the *communication pattern* or *interaction topology* of the complex system.

(2) Parameter space. Simulations deal with the calculation of the time-dynamics of the given complex system for a specific combination of initial parameters. In some cases, the execution of the simulation for a few initial parameter combinations suffices, but typically, a *parameter space search* is necessary to assess the system's behaviour over a range of parameter combinations.

(3) Homogeneity vs heterogeneity. Complex systems can be homogenous or heterogeneous, in that the components may or may not share the same type of state variables. Similarly, the communication pattern can also be non-uniform, when the dependencies of the update functions are heterogeneous. Additional complexity may result from the number of components changing in time, albeit in a theoretical discussion this can always be circumvented by giving an estimate of the maximum number of components. In some cases, the communication pattern is dynamic, even among a static set of components. Formally, this issue can also be avoided by assuming dependence on all other components, some of which dependences may be rendered temporarily inactive. However, our goal is to exploit as much of the available dependency information as possible, therefore, we will explicitly deal with the temporal dynamics of the interaction topology when necessary.

(4) In-run and inter-run parallelization. For exploiting the benefits of a distributed, parallel implementation two major strategies exists. One distributes entire simulation runs across a pool of computers (termed in various ways: *parameter space search*, *parameter sweeping*, *inter-run parallelization*, etc.), while the other attempts to distribute individual runs across computers (sometimes termed *in-run*, or *intra-run parallelization*). It is worth noting that in many cases complex systems simulations are communication heavy, implying that their distribution incurs a strong communication penalty. Yet, in some cases, it is still worthwhile to distribute them for memory gain: a distributed implementation allows for experimenting with large systems (in the number of components) that would otherwise be impossible or very costly.

In the following discussion, we will treat in-run and inter-run parallelization in a unified discussion. Our goal in doing so is to provide a short check-list and a characterization that helps in identifying the potential benefits of a distributed implementation for any given complex systems simulation.

3.1. Partitioning Complex Systems Simulations

The main challenge addressed here is the allocation of complex systems components to computational nodes, i.e., the partitioning of the complex systems simulation, subject to minimizing the execution time of the distributed implementation. (Note, that a version of the minimizing requirement is present also in the case when the goal of distribution is the memory gain.) To cut down on execution time in a distributed environment, one must

- *Minimize communication* among components in different partitions, and
- *Balance the computational load* at each partition (i.e., execution time in between information exchange among the partitions).

In order to achieve the latter, the computational activities at each computational node should be in proportion with the performance of that node. Grid systems are inherently heterogeneous and dynamic and, as discussed earlier, complex systems components can also be heterogeneous and dynamic also. These features taken together would make any abstract analysis very hard. Therefore, we decided to focus on the communication aspect of distribution, trying to classify complex systems simulations based on their communication patterns.

A simple way of putting this is formulating the *sufficient* assumption, that all computational nodes are uniform and homogenous. However, this is not a *necessary* requirement, our classification of communication patterns below can be also used to develop distributed simulations for non-uniform computational systems as well. In fact, more sophisticated implementations based on these classifications are expected to take load-balancing aspects into account as well. On the other hand, in grid systems, communication costs are often prohibiting across computational clusters, while they are more relaxed among computers belonging to the same cluster. Therefore, our focus on minimizing communication is also useful to allocate complex systems components to computational clusters, assuming that load balancing will be handled locally, at the local clusters.

3.2. Communication Templates for Complex Systems Simulations

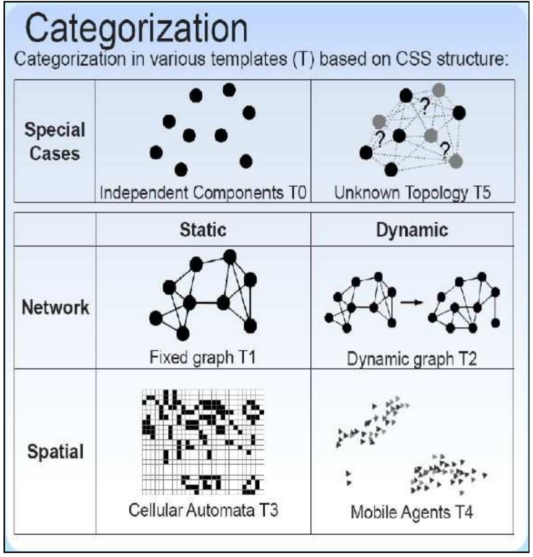
Our approach here is to study the interaction topology of complex systems simulations in isolation, in an attempt to classify frequently occurring cases into communication *templates*. Each identified template can then be accompanied by template implementations and usage advices, based on the classic parallel and distributed computing literature. The general idea here is that complex system modelers can (i) first identify the communication class their models belong to, and then (ii) "fill in" one of the implemented simulation templates provided with their model-specific details. This way, they might not achieve the most efficient distributed implementation of their models, but their implementation efforts will be significantly lowered.

We first point out two extreme cases. In the first scenario, dependencies among components are completely random (e.g., from a uniform random distribution) and change regularly in time (e.g., dependencies are re-sampled prior to each update). The second special case is when no communication occurs among complex systems components. One might argue that these are "useless" or "hopeless" examples, and that a collection of components like in our second example is hardly a *system* at all. However, we believe this is but a question of level of abstraction. In the first scenario, which is not as uncommon as it may seem, there is hardly a better advice regarding a distributed implementation than relying on a distributed parameter space search. Which is just an example of our second scenario: the individual simulation runs can be viewed as non-interacting components. On the other hand, more sophisticated parameter space search methods introduce dependencies among individual runs, by determining the next parameter combinations to explore based on results collected earlier (i.e., sampling in the more turbulent parameter regions, etc.). In this case, the parameter space search becomes a non-trivial complex system again, worthy of dependency analysis on its own.

Dealing with more complex cases, our first observation is that *static* communication patterns allow for the direct application of distribution algorithms. Therefore we will handle these cases separately from the *dynamic topologies*.

Next, we point out that the dependencies of the update functions may be dependent on the components' states. In many cases, the components' state information can be projected to a metric space and update dependencies correlate with distance in this space. For example, if components are agents moving in a space (in computational models often on a two-dimensional lattice) then, each agent's state will (among possibly other things) include the coordinates of the agent. If in the model the agents interact only with the agents in their vicinity, then the update dependencies of the components will be thus distance-dependent. This *spatial property* of a complex systems simulation, if present, may be successfully exploited in determining the partitions of a distributed implementation. It is worth noting, however, that complex systems may have such a *spatial property* implicitly. For example, a social system where people are likely to interact with like-minded partners may have this property where the natural metric space is an abstract similarity space.

Table 1. Categorization of the Various Communication Templates



Based on the observations above, we propose 5+1 *communication templates* that, as we believe, are the commonly occurring classes of complex systems simulations.

Certainly, it would be possible to identify many more, or to refine this classification by dividing some of the templates proposed here. However, we believe these 5+1 patterns[1] are applicable (see Table 1), to a varying extent, a large majority of complex systems simulations, and that useful advices can be formulated for each of them.

Template0 (T0) of our classification is the case where no interaction occurs among components. Here the components' partitioning is only constrained by load balancing considerations.

Template5 (T5), the other extreme, is the case with random or unpredictable interaction among components, for which we suggest 'to step one level up' and the implementation of a distributed parameter search as in Template0. The four remaining cases are created at the intersection of the spatial/non-spatial and static/dynamic properties.

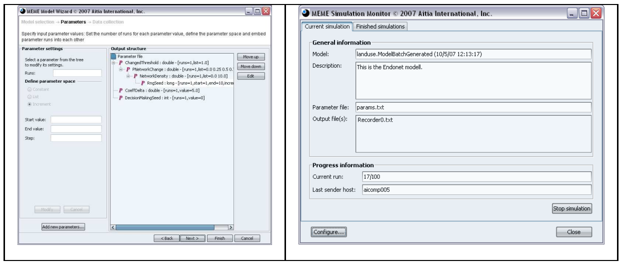
---

[1] We identify 6 communication topologies (or templates), but since for the two extreme cases (T0 and T5) our proposed technical solution is very similar, albeit for different reasons, we prefer to talk about 5+1 templates.

Template1 (T1) (*Static Networks*) describes a non-spatial system with a static communication pattern. It is assumed that the exact communication pattern can be extracted from the system, or that it is defined explicitly. To the thus defined *communication graph* a variety of *graph partitioning algorithms* can be applied. (Fjällström, 1998)

Template2 (T2) (*Dynamic Networks*) introduces dynamism in Template1. The assumption about the existence of a communication *graph* is maintained, but, in contrast to Template5, it is assumed that the changes and their frequency are defined by a graph transition function that provides enough information on the system's efficient distribution. (Sometimes, it may be sufficient to know that the level of change in the communication graph is low, such that it is sufficient to re-partition nodes at every 10000 time steps, or so.) The implementation approach we propose for such systems is the regular application of classic *graph repartitioning algorithms* (i.e., graph partitioning algorithms that attempt to improve on an existing partition). (Barnard and Simon, 1993) Please also note, however, that in the limiting case, Template2 leads to Template5.

Template3 (T3) (*Static Spatial Systems*) moves away from Template1 along the other axis. It maintains the assumption about a static communication pattern, but requires the *spatial property*. Prime examples of such systems are *cellular automata*. Template3 is the pattern most prone to a distributed implementation, most of which can be derived from methods and algorithms developed for distributed cellular automata implementations. (Mazzariol et al, 2000)(Maniatty et al, 1998)


MEME Model Wizard © 2007 Aitia International, Inc.
Model selection → Parameters → Data collection

Specify input parameter values: Set the number of runs for each parameter value, define the parameter space and embed parameter runs into each other

ChangedThreshold : double - [runs=1,list=1.0]
PNetworkChange : double - [runs=1,list=0.0 0.25 0.5 0.
NetworkDensity : double - [runs=1,list=0.0 10.0]
RngSeed : long - [runs=1,start=1,end=10,incre
CoeffDelta : double - [runs=1,value=5.0]
DecisionMakingSeed : int - [runs=1,value=0]

Model: landuse.ModelBatchGenerated (10/5/07 12:13:17)
Description: This is the Endonet modell.
Parameter file: params.txt
Output file(s): Recorder0.txt

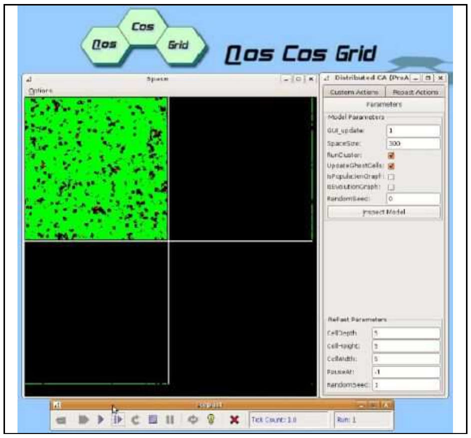
Current run: 17/100
Last sender host: aicomp005

Fig. 1. The Pages from the Distributed Parameter Sweep Wizard for Repast

Finally, **Template4 (T4)** (*Dynamic Spatial Systems*) assumes a spatial system, in which the communication pattern evolves over time. One example of such system is the above discussed case when agents move in space and communicate with those in their vicinity. For the distribution of complex systems belonging to this template, we point to algorithms specially developed for such systems, using buffering and messaging solutions and ways of predicting the speed of spatial movement. (Scheutz and Schermerhorn, 2005)(Gilbert et al, 2006) (The latter may be used to predict the next time communication among partitions or repartitioning becomes necessary.) It is worth pointing out that the fundamental assumption of this template and thus a key to the successful implementation of these solutions is that changes in spatial positions are slow relative to the frequency of state updates.

3.3. Implementation Status

Above we have categorized complex systems simulations into 5+1 classes, based on their internal communication structure. These *communication templates* form the base of the *simulation templates* written in Repast that QosCosGrid will provide for complex systems modelers.



GUI_update: 1
SpaceSize: 300
RunCluster: ☑
UpdateGhostCells: ☑
isPopulationGraph: ☐
isEvolutionGraph: ☐
RandomSeed: 0
CellDepth: 5
CellHeight: 5
CellWidth: 5
PauseAt: -1
RandomSeed: 1
Tick Count: 1.0
Run: 1

Fig. 2. Screenshot from a Distributed Cellular Automata Simulation Implemented with the T3 Simulation Template.

The project reported in this paper is work in progress. A number of working results are already available. An advanced tool for user friendly and extensible parameter sweeping (a solution offered for T0 and T5 above) of Repast simulations with ProActive was reported in (Iványi et al, 2007) and is available from http://meme.aitia.ai/. (See Fig. 1.) Furthermore, simulation templates for models with complying with T1 or T3 were reported in (de Back et al, 2008). (See Fig. 2.) The development of simulation templates for T2 and T4 is an ongoing effort.

4. DISCUSSION AND RELATED WORK

Distributed programming is almost as old as computer science itself. Therefore the problem is *graph partitioning* (which is the theoretical abstraction of communication minimization and load balancing) is an aged and well-studied problem. (Fjällström , 1998) In fact, the simulation templates proposed above for T1-T4 are all based on decades old algorithms, except for one, T4. (Barnard and Simon, 1993) (Hendrickson and Leland, 1995)(Karypis and Kumar, 1997)(Ou and Ranka, 1997)

On the other hand, complex systems simulations have been implemented for supercomputers for several decades now (depending on the exact definition of the field). However, these implementations were mostly done by professional computer scientists and were tailored to the particular problem and infrastructure at hand. (That is, naturally, the approach to take for the most efficient implementation.)

Over the last decade, the increasing capacity of computers and the advent of the agent-based modeling approach made possible to describe models in a way close to the modeled system. This development opened the way for a whole new class of scientists to create and study their models without relying on professional programmers. However, a distributed implementation was so far beyond the reach of these modelers. Work only recently started to address this problem. (Chen et al, 2008) The simulation templates proposed in this paper belong to these novel efforts. They are intended to be used as a blueprint by complex systems modelers.

5. CONCLUSIONS

This paper reported on an ongoing effort to develop templates for distributed simulations on the grid with the integration of the Repast and ProActive packages. The approach to manage the multi-dimensional optimization problem of program partitioning was discussed, together with a classification of complex systems simulations into 5+1 categories, based on their communication structure. These communication templates are then used to develop implementation schemas for complex systems simulations.

The general context of is a *quasi-opportunistic* approach (i) to develop a special purpose middleware that augments grid systems with supercomputer-like properties (number of processing units, temporarily static communication channels, etc.), and (ii) to provide complex systems modelers with a flexible and easy-to-approach platform to develop distributed agent-based simulations.

ACKNOWLEDGMENTS

The present work is being carried out within the frame of the QosCosGrid (Quasi-Opportunistic Supercomputing for Complex Systems Simulations on the Grid) project funded by the European Commission's 6th Framework Programme. The partial support of the EC under grant *QosCosGrid* IST FP6 #033883 is gratefully acknowledged.

REFERENCES

de Back, W., Szemes, G., Kampis, G., Gulyás, L., „Distributed simulation templates for repast", In *Conference of the Swarm Development Group*, Chicago (SwarmFest 2008), 2008.

Barnard, S. T., & Simon, H. D., „A fast multilevel implementation of recursive spectral bisection for partitioning unstructured problems", In *Proc. 6th SIAM Conf. Parallel Processing for Scientic Computing* (pp. 711-718), 1993.

Baude F., Baduel L., Caromel D., Contes A., Huet F., Morel M. and Quilici R., 'Programming, Composing, Deploying for the Grid", in *GRID COMPUTING: Software Environments and Tools*, Jose C. Cunha and Omer F. Rana (Eds), Springer Verlag, January 2006.

Chen, D., Theodoropoulos, G. K., Turner, S. J., Cai, W., Minson, R., and Zhang, Y., „Large scale agent-based simulation on the grid", *Future Generation. Computer Systems,* 24, 7 (Jul. 2008), 658-671.

Coti, C., Herault, T., Peyronnet, S., Rezmerita, A., Cappello, F., „Grid Services For MPI", *Proceedings of the 8th IEEE International Symposium on Cluster Computing and the* Grid *(CCGrid'08), to appear*, Lyon, France, 2008. 05. 14.

Fjällström, P.-O., „Algortihms for Graph Partitioning: A Survey", *Linköping Electronic Articles in Computer and Information Science*, 3, http://www.ep.liu.se/ea/cis/1998/1010/, 1998.

Gilbert, N., den Besten, M., Bontovics, A., Craenen, B. G. W., Divina, F., Eiben, A. E., et al., „Emerging Artificial Societies Through Learning", *Journal of Artificial Societies and Social Simulation* 9 http://jasss.soc.surrey.ac.uk/9/2/ 9.html, 2006.

Hendrickson, B., & Leland, R., „An improved spectral graph partitioning algorithm for mapping parallel computations", *SIAM J. Sci. Comput.*, 16, 452-469., 1995.

Iványi, M., Gulyás, L., Bocsi, R., Szemes, G., Mészáros, R., „Model Exporation Module", *Agent 2007: Complex Interaction and Social Emergence Conference*, Evanston, IL, November 15-18, 2007

Karypis, G., & Kumar, V.. „A coarse grain parallel formulation of multilevel k-way graph partitioning algorithm", In *Proc. Eighth SIAM Conference on Parallel Processing for Scientic Computing*, 1997.

Maniatty, W. A., Szymanski, B. K., & Caraco, T., „Parallel computing with generalized cellular automata", *Parallel and Distributed Computing Practices*, 1, 31-50., 1998.

Mazzariol, M., Gennart, B. A., & Hersch, R. D., „Dynamic load balancing of parallel cellular automata". Paper presented at the *SPIE Conference: Parallel and Distributed Methods for Image Processing IV*, San Diego, USA, 2000.

North, M.J., Macal, C.M., *Managing Business Complexity: Discovering Strategic Solutions with Agent-Based Modeling and Simulation* (Oxford 2007)

North, M.J., N.T. Collier, and J.R. Vos, "Experiences Creating Three Implementations of the Repast Agent Modeling Toolkit", *ACM Transactions on Modeling and Computer Simulation*, Vol. 16, Issue 1, pp. 1-25, ACM, New York, New York, USA (January 2006).

Ou, C.-W., & Ranka, S., „Parallel incremental graph partitioning", *IEEE Transactions on Parallel and Distributed Systems*, 8, 884-896, 1997.

Scheutz, M., & Schermerhorn, P.. "Adaptive Algorithms for the Dynamic Distribution and Parallel Execution of Agent-Based Models". *Journal of Parallel and Distributed Computing*, 66, 1037-1051., 2005.

On-board Operative Adviced Expert Systems for One-Seat Aircrafts and Structure of their knowledge Bases.

Boris Evgenjevich Fedunov
State Research Institute of Aviation Systems (GosNIIAS).
Victorenko st. 7, Moscow 125319, Russia
tel + (095)157-93-49, fax + (095)157-75-13
e-mail: boris_fed@ gosniias.su.

ABSTRACT

The general semantical structure of an on-board real-time advisory systems for a choice of the manner of the attainment of the purpose (" R-T-AS for CMAP") is obtained. The knowledge base of the " R-T-AS for CMAP" consist of the productional rules of the activization of the scenario; mathematical models of the important events; the productional rules of the each scenario. It is showing to resulting of this method. The example of the " R-T-AS for CMAP " for a one-seat aircraft was given.

I. INTRODUCTION

The current function purpose is operationally choiced on every complex anthropocentral object (Anth/object). This is the first global level of control (I GLC) on the Anth/object. The rational manner of the goal achievement is looked for Anth/object. This is the second global level of control (II GLC). The I GLC tasks and II GLC one are most more the part of the Anth/object work. This part of the work is less of all dispatched with the on-board computer algorithm now.

The II GLC tasks are going to decide yet "tomorrow". These dicisions are on-board real-time advisory systems for a choice of the manner of the attainment of the purpose (" R-T-AS for CMAP") for typical situations (TS) of the Anth/object function.

The semantical classifications of an Anth/object is given in [1,2]. The " R-T-AS for CMAP " embraces the certain functi -on-clouse field of the Anth/object function which is named by TS.

II. INTELLIGENT SYSTEMS FOR ANTHROPOCENTRAL OBJECTS.

Intelligent systems attract the attention of the users and designer because of the opportunity to improve the functioning quality of the systems operator-object. This is declared in publications and shown in demonstration and test specimens of some intelligent systems. Such systems allow the accumulation, reproducion, and use of the domain knowledge. The change-over from data to knowledge is a natural result of development and complication of on-board information systems.

Let us turn our attention to intelligent systems that are under development in aviation.

Off-board intelligent systems of preparing an operational flight of an aircraft make the crew ready to execute a particular flight mission. These intelligent systems provide

a) analysis of a priori reconnaissance data about the flight area; in particular, analysis of expected counteraction of air and ground enemy;
b) development of optimal routes of flight in a battle area and returning to the airdrome;
c) choice of variants of tactics of arrival at the battle area and that of fighting against the expected enemy;
d) preparation of flight documents and input data required on board (for airborne equipment and a pilot map).

Note that, at present, all this work is always done before the flight; however, the quality of the preparation (which includes a degree of detail and cordination of flight missions for every member of crew, the supervision of understanding of fligth tasks by the members of the crew, and the accuracy and timeliness of preparation of the corresponding documentation) with the use of intelligent systems is by far higher.

It should be emphasized that an intelligent system is not aimed at doing something new that the operator does not usually do. It does those thihgs that the operator must do, but sometimes fails to do or does badly for various reasons. In addition to this, it always maintains the required technology of preparation and implements all necessary works.

The second group of intelligent systems includes on-board real-time advisory expert systems (“R-T-AS “) and expert systems of on-board measuring and executive devices. We will briefly characterize these systems. On-board real-time advisory expert systems belong to the class of so-called hybrid real-time expret systems. The aim of such systems is to make recommendations to the operators of man-machine objects as to how to solve the problems that face the object.

An aircraft falls into the class of complex antropocentral objects. The problems that face a complex antropocentral object can be divided into the three following global control levels (GCL) [1]:

(1) formulation (choice) of purpose of functioning of the object;
(2) choice of the method of attainment of the purpose formulated on the first GCL;

(3) realizition of the method chosen on the second GCL.

These problems are solved jointly by the operator (his actions are determined by the instruction and are supported by the information control field of the cabin) and by on-board software. Only the third global control level is hardware- and software-supported in the existing systems (in the table, such systems are denoted by the term “today”). In the developments that will arise in the nearest future (in the table, the term “tomorrow” is used for such developments), partial hardware and software support will appear on the second level, though not for all problems that are solved on this level.

Why are the problems of hardware and software supporting the problems of the first GCL and a part of the problems of the second GCL not solved today nor tomorrow? The answer to this question, in our opinion, is as follows. By using the traditional approach to building on-board software and hardware, the designer has run into the obstacles that cannot be overcome with the use of such an approach. These obstacles are the poor structuring of the problems of these levels; dissimilarity of the information required for solving fhese problems with regard to its quality, completeness, and hardware accessibility; a large body of the information about the conditions and general ways of functioning of the system operator-object (fundamental knowledge about the “world”).

Is it possible to overcome these difficulties by means of “R-T-AS “s? The answer is yes, especially for problems of the second GCL.

The hardware and software support of the activities of the crew on these GCLs is presented in the table.

The on-board real-time advisory expert systens of the kind discussed here are designed mainly for the problems of the second GCL, where “ R-T-AS for CMAP “ must work.

Expert systems of on-board measuring and executive devices (ES of OBMED) provide the most complete information about the environment and the state of the on-board hardware required at the moment, and guarantee the most exact execution of the decisions made. They work closely with the “ R-T-AS FOR CMAP “.

The off-board intelligent system of analysis of the results of the use of the anthropocentral object obtains the information from the on-board system of the unbiased control and from the built-in control system, and then determines, together with the crew, the quality of functioning of the system “operator-on-board hard-ware” (the system “pilot-on-board hardware”, in aviation) and effect on the efficiency of the flight.

The off-board intelligent system of diagnostics of the on-board hardware obtains information from the on-board system of the unbiased control, from the built-in control system, and from the standard monitoring-recording hardware. Based on this information, it provides, together with the technical personnel, the analysis of functioning of the on-board hardware in flight, isolation of the faults, and determines how to remove them.

III. ON-BOARD REAL-TIME ADVISORY EXPERT SYSTEMS AND THEIR FEATURES.

(1) The on-board real-time advisory expert systems are designed for joint work partly with the operator on the first global level and, mainly, on the second global level. In the system design [2-4] og the on-board software and indication hardware, these levels correspond to those called “Choice of a typical situation (TS)/typical battle situation (TBS)” and “Decision-making in subsituations with the chosen TS/TBS”, respectively. Note once more that the present-day on-board software and indication hardware is used only on the third GCL (the level of implementation of the decision taken). The on-board real-time advisory expert system of practical significance musbe in agreement with the current conceptual model of the operator behavior and have imperceptible reaction time for the operator (compared to time characteristics of real changes in the environment and those of the activities of the operator).

While the second requirement is accepted by the designers as a natural and concrete one, the first requirement requires discussion. The operator activity incorporates timely and correct detection and understanding of a problem, search for possible ways of solving it, selection of the most judicious (optimal) way, implementation of the solution, and control of the results of the activity. Note that neither the lack of necessary information nor the shortage og the time for its analysis relieves the operator of the necessity of making a particular (better, optimal) decision by some definite instant of time, which is determined by current conditions. It is under these circumstances that the “ R-T-AS FOR CMAP “ must give recommendations to the operator on how to solve a problem that faces him. In addition to this, one should take into account that the technical possibility of interaction between the operator and “ R-T-AS FOR CMAP “ on board is limited, and that the body of aprioriand current qualitative and quantitative information at the disposal of the operator is very large. With this in mind, let us make the first of the above requirements more specific.

To satisfy this requirement, the designer of the “ R-T-AS FOR CMAP “ should take into account the following:

(1) The operator plays the main part on board, and he must not inform the “ R-T-AS FOR CMAP “ about his current plans and ask the recommendations required at the moment. In other words, the knowledge base (KB) of the “ R-T-AS FOR CMAP “ and its conclusion mechanisms must detect and present to the operator significant (in the current conceptual behavior model) events, interpret them correctly, and make recommendations on how to solve the problem, obtained as a result of in-depth analysis.

(2) For every situation significant to the operator that may arise in the context of the conceptual model initiated by the operator, the “ R-T-AS FOR CMAP “ must give convincing and constructive recommendations. In other words, the subject domain of the “ R-T-AS FOR CMAP “ must be functionally closed for the operator, too.
(3) The “ R-T-AS FOR CMAP “ must be semantically and informationally built in the real (under design) information control field of the cabin. In other words, the recommendations and comments to them must be presented in such a form and place that are natural for a particular work station of the operator and be built in the natural space-time world of the cabin.
(4) The direct regime of communication of the operator with the “ R-T-AS FOR CMAP “ is very limited by hardware conveniences of modern cabins and by strong time limitations.
(5) Every particular copy of the “ R-T-AS FOR CMAP “ will be sequentially used by a few operators, who differ from each other by their professional training, psychophysiological cast, and motivation level.

IV. CONCEPTUALIZATION OF THE SUBJECT DOMAIN FOR A TYPICAL SITUATION. STRUCTURE OF “ R-T-AS FOR CMAP “.

The conceptualization of the subject domain for development of the “ R-T-AS FOR CMAP “ means the process and the result of creation of such a formal model the subject domain that (1) would correctly represent a collection of the objects, motivation, the way and result of their functioning; and (2) would allow the system programmer to develop the software based on this model.

Before proceeding to the conceptualization of the subject domain, one should

(1) extract some domain of functioning of the future “ R-T-AS FOR CMAP “ which is functionally closed for the operator;
(2) develop for this domain a generalized graph of activities of fhe system “operator-object-functioning domain”;
(3) outline an available level of the hardware used for the interaction of the operator with the “ R-T-AS FOR CMAP “:
(4) before the functioning of the object (e.g., before the flight) and (b) in process of the object functioning (in-flight conditions);
(5) outline a possible (i.e., hardware acceptable) mechanism of improving the knowledge base of the “ R-T-AS FOR CMAP “ in the process og its functioning.

Such a description of the subject domain is given in a natural professional language; the body of the description must be enough to make the situation clear for the system engineer.

Proceeding to the process of the conceptualization of the aviation subject domain, note that professional pilots always get ready for a flight by carefully clarifying the aim of the flight and expected flight conditions (both favorable, such as external information support and the aid of another aircraft, and unfavorable, such as counteracting objects and bad meteorological conditions). They think of the flight as a number of typical situation (TS) ordered by causal relations.

It should be noted that the preparation procedure itself, regulated and supported by the corresponding technical documents (e.g., the field manual and directions for use), makes the problem and flight conditions well-structured. This structurization and preflight information tactical preparation of the crew must be presented in the knowledge base of the “ R-T-AS FOR CMAP “.

For our purposes, the notion of typical situation (TS) is important. By this we mean a functionally closed part of the work with an explicitly formulated purpose implemented by the system “pilot-on-board hardware-aircraft”. The TS occurs in various possible (real) flights, taking a concrete form in a particular flight. The set of TSs for every aircraft type consists of a minimum necessary number of elements that are required to represent any flight mission.

It seems likely that the case of a completely intelligent object, there will exist a separate “R-T-AS FOR CMAP “ for every TS.

A formal description of a TS written with a natural professional language must contain the following:

(1) The conditions of occurrence of the TS.
(2) The main purpose of the TS.
(3) The performance index and admissible ways of attainment of the purpose of the TS.
(4) Representation of the TS as a set of subsituations ordered according to the relationship of cause and effect.
(5) Participants of the TS and information about them. Purposes of the participants in this TS and the ways of realization of these purposes.
(6) Partners, opponents, external information support, and their general characteristics from the standpoint of attainment of the purpose of functioning of the object.

Note that the representation of the TS through the ordered set of subsituations outline the strategy of attainment of the purpose of the TS itself. The formal description of the subsituations is made as an elaboration of the corresponding part of the formal description of the TS. The formal descriptions of the TS and subsituations are accompanied by a glossary of notions and relations between them, which is necessary for the further development of logical lingustic models [8] for each subsituation and for the whole TS. Let us illustrate this by the example of an aircraft.

The components of the “ R-T-AS FOR CMAP “ for a fighter aircraft are presented in Fig. 1 (only those TSs that will be mentioned below are depicted in the figure).

In the first place, our aim is introduce elements of airficial intelligence and expert systems into the levels “choice of the way of attainment of the purpose taken” (the level of the TS) and “realization of the way of attainment of the purpose taken” (the level of the TS) and “realization of the way of decision-making in the current subsituation”. By that moment, the following typical battle situations (TBS) are best understood and most ready for development of the “ R-T-AS FOR CMAP “ (that is why they are chosen for representation in Fig. 1).

“Throwing the group into a battle” (TGB): (1) with air targets and (2) with surface targets.

“Long-range rocket battle (an attack against one air target)” (LBA-1).

“Long-range rocket battle (an attack against N air targets)” (LBA-N).

“Long-range rocket battle (an attack against surface targets)” (LBS).

For an antropocentral object, every typical battle situation will be “serviced” by its own “ R-T-AS FOR CMAP “ and expert system of on-board measuring and executive devices.

Let us consider the methodology of development of the knowledge base of the “ R-T-AS FOR CMAP “ for some TS. For each subsituation of this TS, let us make a list of objects-participants and significant events. Let us represent each subsituation of this TS by a set of mathematical models (MM), which describes the space disposition of the participants of the subsituation, predict its change in time, and determine possible moments of occurrences of the events which are significant for the subsituation under consideration. This set will be referred to as the scenario corresponding to the subsituation. The subsituation often needs some preliminary investigations on a number of mathematical models, which, as a rule, are formulated in the form of optimal control problems, problems of game theory, and various decision-making problems [9]. Some “judicious” solutions for this subsituation, which was obtained as a result of these investigations or simulation modeling, is used in the mathematical model. The union of the mathematical models, together with a reasonable (with respect to the performance index) behavior of the object carrying the “ R-T-AS FOR CMAP “ and objects-opponents, form the space-time framework of the scenario. Scenarios of the subsituation are related to each other in the “ R-T-AS FOR CMAP “ by such a causal relation that allows the description of the proceeding of the TS in the varying environment by switching from one scenario to another.

Analysis of on-board and off-board conditions, as well as making of the corresponding recommendations to the operator, will be done by means of production rules. A set of such rules for the “ R-T-AS FOR CMAP “ are included in its scenario. The rules of scenario initiation are placed in a separate block.

Functional blocks of the “ R-T-AS FOR CMAP “ are presented in Fig. 2. These blocks contain the knowledge necessary for the functioning of the “ R-T-AS FOR CMAP “ given in the form of production rules and mathematical models.

Figure 2 illustrates how the “ R-T-AS FOR CMAP “ of a typical situation is related to the information control field of the operator cabin and expert systems of on-board measuring and executive devices.

Let us sum up the above discussions.

(1) The “ R-T-AS FOR CMAP “ should be built for a functionally closed part of the work of the system “operator-on-board hardware”. The process of formalization of the subjec domain is divided explicitly into two stages. At the first stage, a semantic net of frames and a set of mathematical models of the subject domain, which represent its space-time “world”, are built. The substages of this stage include the description of the subject domain in a natural professional language with the subsequent change-over to protolanguages, making a generalized graph of the functioning of the system “operator-on-board hardware”, development of a semantic net of frames, and determination of the necessary collection of mathematical models.

At the second stage, a logical linguistic model of the subject domain is built, and hierarchically ordered sets of inference rules and mathematical models are developed.

To design the “R-T-AS FOR CMAP “of practical significance, such descriptions must be specified, and their completeness and consistency should be supervised.

(2) In preparing the anthropocentral object for use (in a flight), the crew think of the mission as a sequence of typical situations (TS) related to each other by the relationship of cause and effect.

Any mission can be represented as a set of such typical situations. We think of the TS as a part of the functionally closed work of the system “operator-on-board hardware”, for which the “R-T-AS FOR CMAP “ is developed. There must be a set of “R-T-AS FOR CMAP “s of TS on board. A particular OBRATES of TS is activated by the crew (operator) or by a special OBRATES of the first global level (“R-T-AS FOR CMAP“ of GCL1).

The process of attainment of the purpose in a chosen TS is represented naturally as a sequence of subsituations-scenarios of the “ R-T-AS FOR CMAP “. Mathematical models and a system of production rules are grouped in the “ R-T-AS FOR CMAP “ according to the scenarios. The rules of initiation of a particular scenario are contained in a separate initiation block.

The mechanisms of the conclusion in knowledgebase on-board operative advising expert system are given in [4,5].

TABLE I
THE HARDWARE AND SOFTWARE SUPPORT OF THE CREW ACTIVITIES ON THE FIRST, SECOND AND THIRD GLOBAL CONTROL LEVELS (GCL)

Global con-trol levels for antropo-central object	Hardware and software on-board facilities	
	“tomorrow ”	“the day after tomorrow”
1st GCL Choice of the purpose	“ “...”	“_>A&DS_” “R-T-AS” “Choice of a TS”
2nd GCL Choice of the way of attain-ment of the purpose	“_>>A&DS_” “...”	“_= A&DS_” “ R-T-AS FOR CMAP “
3d GCL Inplementation of the chosen way	“_= A&DS_” “...”	“_= A&DS_” ES OBMED

Notation:

“_“ - no support is available;
“-A&DS” - support by standard an algorithms and display support (A&DS)
“>A&DS” - partial support by A&DS
“>>A&DS” - almost complete absence of the support by the standard A&DS
“...” - there exists a possibility of development of an “ R-T-AS FOR CMAP “ of TS and an expert system (ES) of on-board measuring and executive devices (OBMED).

Summing up the above-stated, we introduce into practice a registration certificate of the common semantic structure of the knowledge base of the real-time advisory system (R-T-AS) for a choice of the manner of the attainment of the purpose (CMAP). It presents on the table 2. Values of the « Duel» are given in the right column of the table 2 as a example.

Table 2.
THE REGISTRATION CERTIFICATE

Description	Value (number for the Duel)
A rule base : • structure of the main hierarchy levels on the rules set , • quantity of rules on each level	1+5 20+(from 20 to 200) x 5
Quantity of the important events Quantity of the mathematical models of the fragments of the problem situation	6 7
Quantity of the independent decides	6+1

In work [3,6-10].they are given fragments knowledgebase on-board operative advising expert systems and some results of modeling.

V.Conclusion.

Present of the on-board place and the structure of the on-board real-time advisory systems for a chice of manner of attainmment of the purpose. The examples of this systems for a one-seat aircraft are discribed in [10 -12].

VI. REFERENCES

[1] Fedunov, B.E. “Problems of the Development of On-Board Real-Time Advisory Expert Systems for Anthropocentral Objects”, Journal of computer and systems sciences international (A Journal of Optimization and Control). ISSN 1064-2307. English Translation of Izvestiya of Rossiiskoi Akademii Nauk. Teoria i Sistemy Upravleniya. Russian Academy of Sciences. Izv. Russ. Akad. Nauk, Teor. Sist.Upr., 1996. № 5. pp.147-159/

[2] Moiseev, N.N., Control Theory and Problems “Man-Environment”, Vestn.Akad.Nauk SSSR, 1980, no. 1.

[3] Fedunov, B.E., The Optimization Models for Taking of the Decisions in the Algorithmic and Indicational Support System Designing SAMS, Berlin, 1995, vol. 18/19.

[4] Fedunov, B.E. Inference technique based on precedents in knowledge bases of intelligence systems

EMSS-207 paper. Bergeggy, Italy,2007

[5] Fedunov B.E. The mechanisms of the conclusion in knowledgebase on-board operative advising expert system. Journal of computer and systems sciences international (A Journal of Optimization and Control). ISSN 1064-2307. English Translation of Izvestiya of *Rossiiskoi Akademii Nauk. Teoria i Sistemy Upravleniya. Russian Academy of Sciences. 2002*, №. 4.

[6] Vasil’ev S.N., Gherlov A.K., Fedosov E.A., Fedunov B.E. Intelligent control of dynamic systems. Moscow, Fizmatlit, 2000, (by Russian).

[7] Fedunov B.E., Tichenko U.E. Optimum moments of the starting the rockets and using the hindrances in duel of the situations aircraft. Journal of computer and systems sciences international (A Journal of Optimization and Control). ISSN 1064-2307. English Translation of Izvestiya of *Rossiiskoi Akademii Nauk. Teoria i Sistemy Upravleniya. Russian Academy of Sciences. 2006*, №. 5. pp. 98-109.

[8] Romanova, V.D., Fedunov, B.E., and Yunevich, N.D., Research Prototype og the OBRTAES “Duel”, Izv. Ross. Akad. Nauk, Teor. Sist. Upr., 1995, no. 5.

[9] Shinar, J., Siegel, A. and Gold, Y., “A Medium-Range Air Combat Game Solution by a Pilot Advisory System”, AIAA 89-3630-CP, Guidance, Navigation and Control Conference, Boston, MA. August, 1989, pp. 1653-1663.

[10] Yaakov, Y., Shinar, J., Wolfshtein, M., and Boneh, A., “A new design approach for a pilot advisory system in air-to-air interceptions”, Proceeding of ICAS-96, Italy, 1996.

AUTHOR BIOGRAPHIES

BORIS E. FEDUNOV, doctor of technical sciences, professor, graduated the Moscow Aviation Institute (State University of Aerospace Technologies) in 1960 as aircraft designing engineer, Moscow State University in 1965 as mathematician. Now he works at the State Research Institute of Aviation Systems and delivers a lectures in Moscow Aviation Institute on syntheses of on-board algorithms systems of piloting aircrafts and applied systems analysis. He is member scientific council of the Russian Association of an Artificial Intellect (collective member of European Coordination Committee on the Artificial Intellect).

MODELING AND SIMULATION OF A NANOSTRUCTURE FOR A SINGLE ELECTRON TECHNOLOGY IMPLEMENTATION

C. Ravariu[(a)], A. Rusu[(a)], A. Bondarciuc[(b)] F. Ravariu[(c)], T. Niculiu[(a)], F. Babarada[(a)], V. Bondarciuc[(b)]

[(a)] "Politehnica" University of Bucharest
[(b)] Spectrum UIF Bucharest
[(c)] IMT-Bucharest

[(a)]cr682003@yahoo.com, [(b)]ala_spectrum@yahoo.com, [(c)]florina.ravariu@imt.ro

ABSTRACT

The Electronic Devices Simulators are valuable tools used in micro and nano-electronics labs. This paper analyses the electrical characteristics evolution, under the down-scaling sizes tendency of the SOI devices. A nanostructure sub-10nm Si film thickness with a vacuum cavity in the device body was simulated. The global current is a superposition of a tunnel current through the cavity and an inversion current at the film bottom. The tunnel source-drain current prevails in devices with sub 10nm film thickness and provides the I_D-V_{DS} characteristics with a minimum. For film thickness comprised between 200-10nm, the I_D-V_{GS} curves preserve similar shapes with a classical MOS/ SOI's transfer characteristics. For sub 10nm film thickness, the shape of the I_D-V_{GS} characteristics tends to have a maximum, like in Single Electron Device SED.

Keywords: Few Electrons Transistors, SOI structures, simulations of nano-devices.

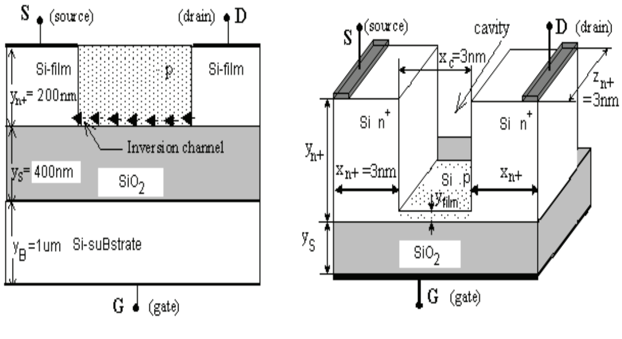
1. INTRODUCTION

The approach of the nowadays micro and nano-electronics themes, at laboratory level in a university without a technological park, is possible just with strong simulators.

In this paper, the some virtual measurements of the SOI transistors with nano-metric films thicknesses are presented. The reference device was a standard SOI-MOSFET, 200nm Si-film on 400nm Oxide (Ravariu and Rusu 2006), fig.1.a. The I_D (V_{DS}, V_{GS}) curves were studied for thinner films: 10nm, 1nm, 0.3nm. For sub-10nm film thickness, a new structure with a nano-cavity, was proposed in fig.1.b. The source and drain regions consist in two high "undulations" of Si-n^+ (y_{n+}=7nm) onto an oxide support. The Si-p transistor body was thinned firstly to y_{film}=1nm and secondly to y_{film}=0.3nm. The carriers transport was confined at the limit, one by one. Essentially, the device could be regarded as a string of "Few Electron Transistors" that converges to the "Single Electron Transistor" (SET), as an ideal limit. Another reason for two prominent n^+ - undulations is related to the practical possibility of Source and Drain metallization. On the other hand, both thick parallelepiped n^+ - layers fulfill the electron reservoir role, like in a SET, (Mahapatra, Ionescu, and Banerjee 2002).

2. THE NOVEL DEVICES ARCHITECURE

The device architecture, presented in fig.1.b, was inspired from a real sub-10nm undulated polysilicon film, (Badila, Ecoffey, Bouvet, and Ionescu 2003). Secco etching in $K_2Cr_2O_7$:HF solution, which currently is applied for the crystalline defects revealing in silicon, preferentially etches the boundaries of the poly-silicon grains, producing undulated poly-silicon layer with maximum 6 nm and minimum 3 nm thickness. The thin undulated poly-silicon films are deposited onto an insulator material. The electrical stimulus were in the range: -3 ÷ +3V, -10V ÷ +10V, -15V ÷ +15V, (Eccofey, Bouvet, Fazan, Tringe, and Ionescu 2003).



(a) (b)

Figure 1: (a) The current flow in a cross-section of a 200nm SOIMOSFET; (b) the nanotransistor with a cavity

Starting from these experimental data, we developed a new device configuration with the source and drain regions are n^+-type silicon ($N_D=10^{17}cm^{-3}$) with x_{n+}=3nm, y_{n+}=7nm, z_{n+}=6nm, placed at x_c =3nm distance. A thinner p – type Si film ($N_A=5\cdot10^{15}cm^{-3}$) links the source and drain regions and represents the inversion channel location. All these parameters will be maintained constant during the simulations. Thinning the p-type film to y_{film}=1nm, a cavity carried out between source and drain.

3. THE ANALYTICAL MODEL

For small vacuum distance d, the source-drain tunneling probability is described by the Fowler-Nordheim model, (Rusu, 2000):

$$P_t \approx exp\left[-\frac{4\sqrt{2m_n^*}\cdot\chi_S^{3/2}\cdot d}{3q\hbar V_{DS}}\right]. \tag{1}$$

where m_n^* is the electron effective mass, χ_S is the semiconductor affinity for electrons in respect with the vacuum, ħ=h/2π (h is the Planck's constant), q is the elementary electric charge.

The tunnel effect through a triangle potential barrier is the main phenomenon in the proposed SOI transistor with a cavity. Some electrons tunnel the Si - Vacuum barrier, producing the tunnel current, I_t , (Rusu, 2000):

$$I_t = \frac{A}{\sqrt{E_b}}\cdot\left(\frac{V_{DS}}{x_c}\right)^2\cdot\exp\left(\frac{B\cdot E_b^{3/2}\cdot x_c}{V_{DS}}\right). \tag{2}$$

where $E_b = \chi_{semic} - \chi_{vacuum}$ is the height of the triangle barrier of the potential from semiconductor to vacuum and A, B are some material parameters depending on the effectiveness mass for electrons and holes.

Considering x_C as constant parameter, the variation of the tunnel current I_t, versus the drain-source voltage V_{DS} is analytically studied via the first order derivative. Because the function $I_t'(V_{DS})$ is positive for $V_{DS}>0$, the tunnel current monotonically increases with the drain-source voltage.

Zeroing the first order derivative of the model (1) results a minimum for the tunnel current versus V_{DS} voltage:

$$V_{DS}\Big|_{I_t=\min} = \frac{B\cdot E_b^{3/2}\cdot x_c}{2}. \tag{3}$$

The simulations proved that the total current preserve the curvature with a minimum when the film thickness decreased under 1nm because the percentage of the tunnel current, I_t overcomes that from the inversion channel, I_{MOS}. For thicker Si-p film (y_{film}>10nm), the tunnel current is negligible and the characteristics tends to those of the classical SOI-MOSFET. The cavity itself has a high vacuum. The number of air molecules N, in the cavity volume for y_{film}=1nm, is:

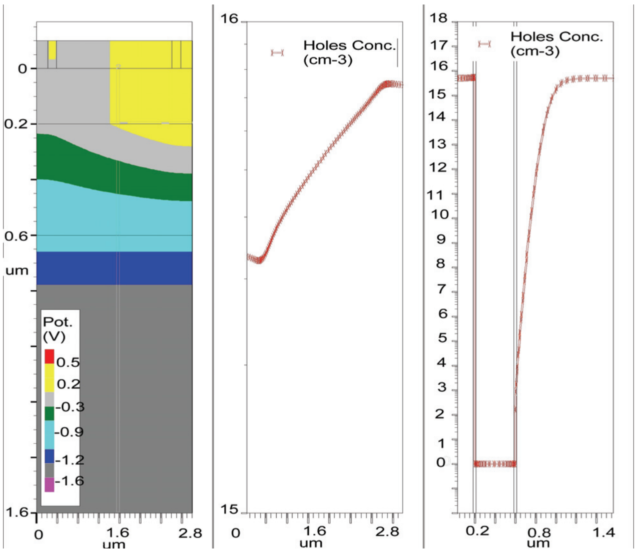
$$N = \frac{x_c\cdot(y_{n+} - y_{film})\cdot z_{n+}\cdot N_{A0}}{V_{m0}}. \tag{4}$$

That means N ≈ 2 air molecules. Hence, the electrons will not be disturbed by the air molecules from cavity, in normal conditions (N_{A0} = 6,023×10^{23} molecules/mol, V_{m0} = 22,42 dm^3/mol). Consequently, the device manufacturing doesn't require a special vacuum technology.

4. THE SIMULATION RESULTS FOR THE SOI STRUCTURE WITHOUT CAVITY

In the ATLAS simulations, the constructive data of the nanotransistor were: y_{n+}=0.2μm, y_S=0.4μm and y_B=1μm, the doping concentrations in film and substrate N_A=2×10^{15}cm^{-3}. The applied voltages were V_G= 0V… -3V, V_D=0V… +4V, V_S=0V.

Figure 2.a presents the potential distribution and the holes concentration in an intermediate situation at V_G = -1,8V, through the structure with 200nm Si-film thickness. A negative gate bias induces a holes crowding in the p-type film. Near drain, where V_{GD} is higher than V_{GS}, the holes reached p=8x10^{15}cm^{-3} and near source p=4x10^{15}cm^{-3} > 2x10^{15}cm^{-3} = N_A, fig.2.b. From the longitudinal holes distribution can be observed the holes concentration decreasing in the substrate, fig.2.c. This simulation proves the substrate depletion effect.



(a) (b) (c)

Figure 2: (a) The potential distribution in the 200nm SOI structure; (b) the holes concentration accros the structure; (c) the holes concentration along the structure

Figure 3 presents the electron concentration in the structure with 200nm film thickness. A positive gate bias induces an electron inversion channel in p-type film (e.g. $n\big|_{y=0.2um} = 10^{16}cm^{-3} >$ 5·10^{15}cm^{-3}=$N_{A\text{-}film}$), figure 3.

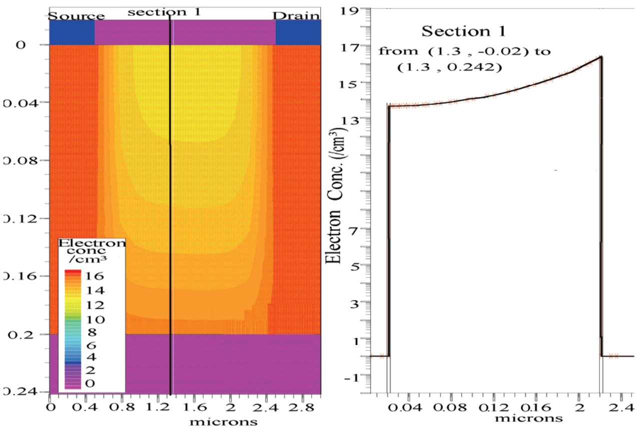


Figure 3: Detail in film for the electron concentration

5. THE SIMULATION RESULTS FOR THE SOI STRUCTURE WITH A NANO-CAVITY

In the simulations, the constructive data were those described in the paragraph 2.

The main physical effects were included as "nano-effects": Band to Band Tunnelling, Fowler-Nordheim tunnelling, Fermi distribution, including in the MODEL statement the following parameters: BBT, FNORD, FERMI. Figure 4 presents the total current vectors in 1 nm-film at V_{DS}=4V, V_{GS}=3V, besides to the electrons concentration in the channel region. The vectors through the vacuum (emphasised by black line), proved the tunnel effect. A value about 10^{20}cm^{-3} means 1 electron/channel, which shows our target: Single Electron Working regime for nanostructure, fig.5.

In fig. 6.a a family of curves I_D-V_{GS} for y_{film}=200nm, 10nm, 1nm, 0.3nm is presented. These curves have a maximum for y_{film}≤1nm, like SET transistor, (Mahapatra, Ionescu, and Banerjee, 2002).

Figure 6.b shows the curves I_D-V_{DS} at V_{GS}=3V. The shape of the I_D-V_{DS} curves with a minimum proves the tunnel effect, accordingly with the equation (1).

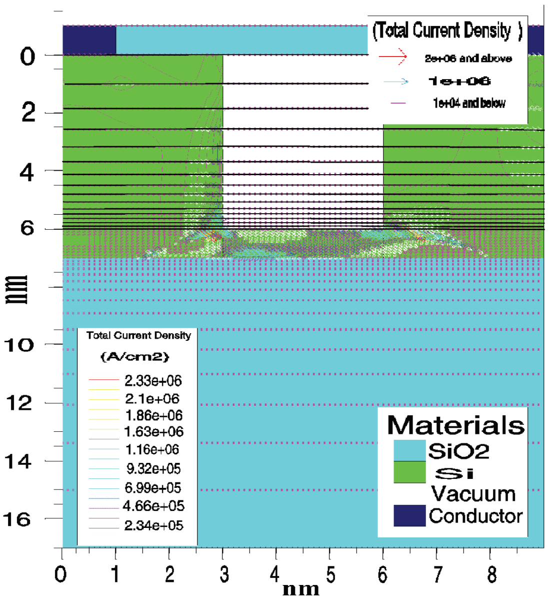


Figure 4: Total current vectors in the 1 nm transistor

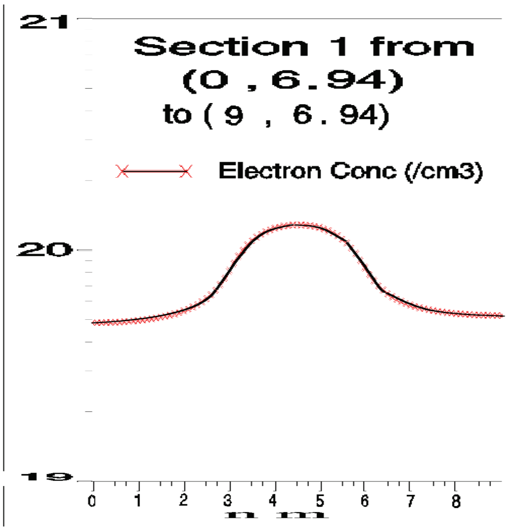
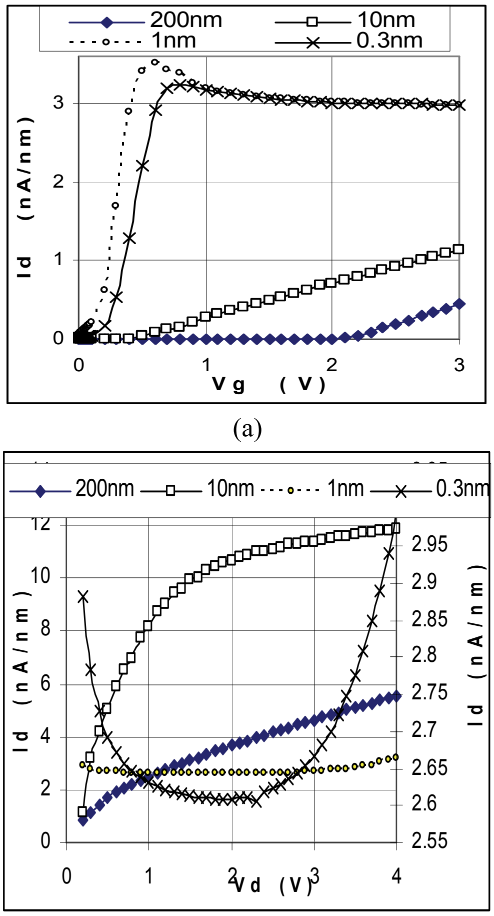


Figure 5: The electron concentrations in the 1nm transistor



(a)

(b)

Figure 6: (a) The I_D-V_{GS} ; (b) the I_D-V_{DS} , characteristics

Figure 7 presents the global potential distribution (left) and a detail of the electron concentration (right) for the 0.3nm structure with cavity, biased at a high drain voltage in this last case (V_S=0V, V_G=3V, V_D=4V).

In this case the saturation occurred and an unbalanced electron distribution can be seen in the film (fig.7): $1.1 \cdot 10^{16}$ cm^{-3} in the source region, $7 \cdot 10^{15} cm^{-3}$ in the channel near the source, $2 \cdot 10^{15} cm^{-3}$ in the channel near the drain and decrease up to $1.4 \cdot 10^{15} cm^{-3}$ in the drain region, at the film bottom.

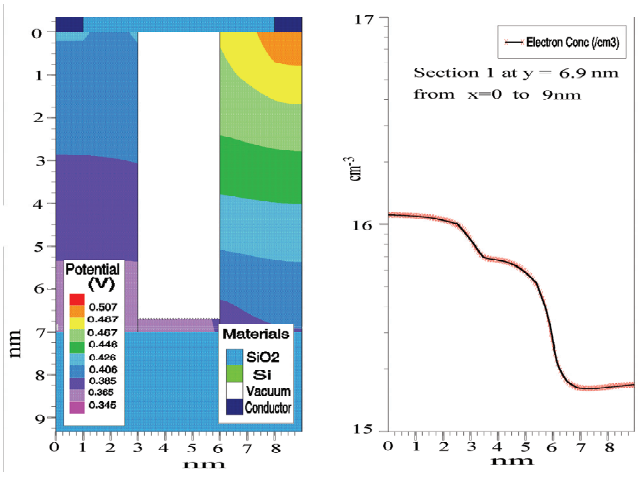


Figure 7: The potential and the electron concentration in the 0.3nm structure

6. CONCLUSIONS

A nanotransistor with Silicon On Insulator structure was presented. When the film thickness varied between 200nm to 10nm the electrical characteristics preserve the classical shape. When the film thickness varied from 1nm to 0.3nm and the cavity occurs above the film, the device presents atypical electrical characteristics I_D-V_{GS}, having a maximum like the SET transistor. The shape of the I_D-V_{DS} curves with a minimum proves the presence of the tunnel effect. The nanocavity comprises 1-3 air molecules in normal conditions, negligible for the current transport. Consequently doesn't imply a special vacuum technology. The electron transport in the p-film is one by one, proving the Single Electron Technology for our proposed SOI nanotransistor.

REFERENCES

Ravariu, C., Rusu, A., Ravariu, F., 2006. Interface electric charge modeling and characterization with<delta>- distribution generator strings in thin SOI films. *Microelectronics Elsevier Journal*, 37(3), p. 943-947.

Mahapatra, S., Ionescu, A.M., Banerjee, 2002. A quasi-analytical SET model for few electron circuit simulation. *IEEE Elec. Dev. Lettere*, 23(6), pp. 366-368.

Badila, D., Ecoffey, S., Bouvet, D., and Ionescu, A.M., 2003. A study of fabrication technique for sub – 10nm thin undulated polysilicon films. *Proceedings of the IEEE Annual Conference of Semiconductors*. pp. 95-98. October 1-2, Sinaia, (Romania).

Eccofey, S., Bouvet, D., Fazan, P., Tringe, J.W., and Ionescu, A.M., 2003. Novel technique for nanograin ultra-thin polysilicon film deposition and implantation. *Proceedings of the IEEE Annual Conference of Semiconductors*. pp. 47-50. October 1-2, Sinaia, (Romania).

Rusu, A., 2000. Non–linear electrical conduction in semiconductor structures. *Edited by Romanian Academy House Publishing*, Bucharest, 140-144.

AUTHORS BIOGRAPHY

Cristian RAVARIU is with the Politehnica University of Bucharest, Romania, Faculty of Electronics Telecommunications and Information Technology, Microelectronics Department, B-dul Iuliu Maniu 1-3, Sector 6, Bucharest, Postal zip: 061071. In the lasts years he studied nano-electronic devices, proposed a Nothing On Insulator configuration, besides to Bioelectronics devices, cellular nano-electronics. e-mail: cravariu@arh.pub.ro

Adrian RUSU is with the Politehnica University of Bucharest, Romania, Faculty of Electronics Telecommunications and Information Technology, Microelectronics Department, B-dul Iuliu Maniu 1-3, Sector 6, Bucharest. He is the head of the Micro-Nano-Electronics Department. His researching area concerns new electronic devices and technologies, non-linear conduction through semiconductors. e-mail: adrianr@mcma.pub.ro .

Ala BONDARCIUC is with the Spectrum UIF private company of Bucharest, Smaranda Braiescu Street, sect.2, Romania, specialist in quantum physics and quantum medicine. e-mail: ala_spectrum@yahoo.com

Florina RAVARIU is with the National Institute for Research and Development in Microtechnologies (IMT Bucharest), Str.Erou Iancu Nicolae 32B, 06996, Bucharest, Romania. She is specialist in micromachining and nanotechnologies. e-mail: florina.ravariu@imt.ro

Tudor NICULIU is with the Politehnica University of Bucharest, Romania, Faculty of Electronics Telecommunications and Information Technology, Microelectronics Department, B-dul Iuliu Maniu 1-3, Sector 6, Bucharest. He is Senior IEEE Member with high performances in artificial intelligence, mathematical algorithms, simulator. e-mail: tniculiu@yahoo.com .

Florin BABARADA is with the Politehnica University of Bucharest, Romania, Faculty of Electronics Telecommunications and Information Technology, Microelectronics Department, B-dul Iuliu Maniu 1-3, Sector 6, Bucharest. He is working in the nanotechnology field, circuit design. e-mail: babflorin@yahoo.com .

Vlad BONDARCIUC is with the Spectrum UIF private company of Bucharest, Smaranda Braiescu Street, sect.2, Romania, specialist in informatics. e-mail: vladutzz@yahoo.com.

SOME ESTHETICAL VALENCES OF THE SPATIAL LOCALITY. TOWARDS ARTIFICIAL ESTHETICS

Cristian Lupu

Romanian Academy
Center for New Electronic Architectures

cristianslupu@clicknet.ro

ABSTRACT

This paper tries to model an "esthetical behavior" supporting the interconnected structures. Based on the concept of structural self-organization it defines the collectivity as a set related by the structural relations. While the structure (esthetical or not) is a concept, the representation or the image is an intuition (according to Croce). Nearby the structure, opposite to the function, is the image as an intuition. The esthetical structures are characterized by significant intuitive representations. Thus, the perception of the structural self-organization of a work of art is, finally, an intuition. Or, the esthetical function is the expression of the work of art. The concept of the esthetical structure and the intuition of the esthetical representation form the two sources of the conception/reception of a work of art. We introduce the notion of "interconnected esthetical collectivity" and, on this background, we try to model an esthetical behavior supporting the esthetical locality concept.

Keywords: interconnected esthetical collectivity, esthetical locality

1. INTRODUCTION

"*The knowledge has two forms: it is an intuitive knowledge or a logical knowledge; a knowledge by the imagination or a knowledge by the intellect; a knowledge of the individual or a knowledge of the universal; of the things considered each separately or a knowledge of their relations; it is, finally, a producer of images or a producer of concepts…. The intuition means, frequently, the perception, i.e. the knowledge of the happened reality, the perception of something as real*" (Croce 1971).

A complex system perception, as of a work of art, means first of all the perception of a *self-organization* of the system or of the *relations* that organize the system. To perceive a complex, said Wittgenstein, *"means to perceive the relations of its constituent parts in a determined way"* (Wittgenstein 1991). On the other hand, one of the natural characteristics is the *association in collectivities*. Professor Moshe Sipper said in the foreword to a recent book, that through the computing terrain during the past few years a new wind has been swept, *"slowly changing our fundamental view of computers. We want them, of course, to be faster, better, more efficient - and proficient – at their tasks. But, more interestingly, we are trying to imbue them with abilities hitherto found only in nature, such as evolution, learning, development, growth, and collectivity"* (Castro and Zuben 2005). We can observe collectivities in the not living world (universe galaxies, solar systems, crystalline units), in the living world (ant hills, bee swarms, nations) as in the artificial world (paintings, especially the abstract ones, architectures, cities). What *properties* are behind the relations that organize the collectivities, or, better said, the *relations of association in collectivities*? Maybe is the gravity, the symmetry or the survival instinct or, maybe, an *esthetical property*? In one word it is *structural self-organization*. The self-organization is based on *structural relations* (not dependable on time) between *structural entities*. The self-organization can be structural or functional (relations dependable on time).

The definition of the term *collectivity* deduces from the definition of the term *set*. "A set can be selected by a *membership* or can be constructed by a *relation which substantiate the membership* or by bringing in the set *elements which fulfill the relation defining it*" (Drăgănescu 1985). Because N. Bourbaki names "*collectivizing relation*" the relation defining a set, we name collectivities only the sets selected or built based on *relations*. Therefore, we exclude the sets selected by the membership, (the most general definition of the set). A collectivity does not mean, in our point of view, a set made, for example, of {*a star, 5, a planet, a crystal, c, an ant, a bee, a man*}.

The relation that proves the membership of a collectivity is resulted from its *structural properties*: a collectivity is made of smaller *structural entities*. For example, an interconnecting relationship is composed of a set of nodes and links which is equivalent with the graph definition (a set X of nodes and an application Γ of X in X which gives the set of connections). The *link*, the *connection* is a *structural property* for an interconnection or a graph.

2. ESTHETICAL STRUCTURE AND INTUITION

A basic concept in this article, which we have used but not explained, is the concept of the *structure*. The *structure concept*, at the beginning with the meaning of building (from the Latin noun *structura*), has slowly advanced. In English and French the word has the same meaning: edifice, way to build. The abstraction of the word makes slowly: only in the XVII-XVIIIth centuries appears in the sense of *a reciprocal relation of the parts* or *the constitutive elements of a whole, determining its nature, its organization* (Nemoianu 1967). During the XIXth century, structure is generally opposite to function, like static to dynamic. The end of the XIXth century brings a new meaning of the structure concept. It will begin to represent not a static organization, but a *whole made by solidary elements, in which everyone depends on all* the other *ones and can be what it is only in and through them*. The *connection between parts* (the first meaning) is something less necessary than the *total interdependence system of each part with all other parts* (the second meaning). If the first meaning is a *sum*, the second is a *whole*. *The whole can dominate the part* (Okakura 2007). This turning point coincides with the penetration of the structure concept in the humanities. The term has been changed by a synonym, *Gestalt*. *Gestalt* is not understood beginning with the form, pattern, structure, but the behavior of an organism, a whole. *Gestalt* is related to an *entelechy*, a term appointing the features of geometric figures or melodies by which they exceed the *sum* characteristics. A geometric figure remains itself even if it is decreased, enlarged, color modified. This invariance of the transposing is also known by the name of *isomorphism*. In this way, the linguistic researchers have contributed resolutely to the understanding and to the using of the structure concept unifying both meanings: the *coherent, coagulated globality* and the *relations system between local parts* or, in two words, the *globality* and the *locality*.

The structure is a concept while the *representation* or the *image* (Croce 1971) is an *intuition*. Nearby the structure, alike opposite to the function, is the image as a intuition. The *esthetical structures* are characterized by *significant* intuitive representations. Thus, the perception of the structural self-organization of a work of art is, finally, an intuition. "*The result of a work of art* (the conception and/or the reception, m.n.) *is an intuition*" (Croce 1971). The representation, in Croce's opinion, is an intuition that detaches and emphasizes on the *psychic background of sensations*. The representation is an *elaboration* of sensations, and therefore is an intuition. The concept of the esthetical structure and the intuition of the esthetical representation (of the image) form, in our opinion, the two sources of the conception/reception of a work of art. By the structure and the intuition, a work of art *closes itself*, the functional behavior isn't necessary to be understood (except the design and the other "functional" works of art). The work of art is a pure structure, an esthetical structure, which must be understood, inferred.

The *esthetical structures* are *esthetical collectivities*, i.e. sets built with the help of the *esthetical relations* resulted from the *esthetical properties*. An esthetical relation is a relation that *spiritually expresses* the connections between the collectivity entities on the basis of the esthetical properties (e.g. synthesized by the binomials *beautiful-ugly* or *symmetric-asymmetric*). The esthetical relations are by definition *structured*. "The complete process of the *esthetical production* can be symbolized in four stages: *a*) impressions; *b*) *expression* or esthetical spiritually synthesis; *c*) hedonic accompaniment or pleasure of beautiful (esthetical pleasure); *d*) translation of esthetical fact in physical phenomena (sounds, tones, motions, combinations of lines and colors). Anybody observes that the essential point, only which is proper esthetical and real is the point *b*, which is absent to the naturalistic manifestation and construction, and which is nominated, at their turn by metaphor, *expression*" (Croce 1971). The (esthetical) expressions are representations or images of an esthetical structure (work of art) which can be perceived in a certain succession, a temporal. The structural self-organization of a work of art means a spiritual esthetical synthesis or an (esthetical) expression. "The esthetical functionality" is replaced by an "esthetical process", the essence of which is, according to Croce, the *expression*.

The *structure of an esthetical collectivity* can be, as any structure, self-organized *locally* and *globally*. An interconnecting structure is locally estimated by neighborhoods. *The locality is the behavior or the structural self-organization of an (esthetical) collectivity around an* ***origin***. In case of an esthetical collectivity, the origin can only be spatial. The article refers to the spatial origins of a structure and the locality definition covers the first meaning of the structure concept (connection between parts). *The globality is the behavior or the structural self-organization of an (esthetical) collectivity around a* ***property***. For example, the works of art can be estimated by the help of *symmetrical* or *asymmetrical* properties. The globality definition refers to the second meaning of the structure concept. Therefore, an esthetical structure can be estimated, as any structure, by measures of the locality and the globality.

On the other hand, the *architecture of an esthetical collectivity*, connection concept between the esthetical structure and the *esthetical function* (the *expression of the work of art*), produces a *global meaning*, an *intuition*, of the collectivity with the aim to understand the unity between the structure and the expression of that esthetical collectivity. We can talk about universe's architecture, a crystallographic system's architecture, a house's or a town's architecture, an enterprise's architecture, a computer's architecture, an interconnecting architecture, a communication architecture or, finally, an esthetical architecture (of a work of art). *The esthetical architecture measures by the degree of membership to certain esthetical global properties*. The symmetry, hierarchy, homogeneity are also global properties, not only esthetical ones. We must not to confound the architecture concept, leading to an intuition on the collectivity, with globality concept, which is a measure of the collectivity.

We shall concentrate, in the following sections of this paper on certain esthetical structures. We try to give examples of works of art (paintings) which we analyze from the point of view of the locality and globality. Analyzing in this way, we probably have succeeded to algorithm a part of the *expressions* of the works of art and to understand *inferring* them. Our application can lead towards an "*artificial*" *esthetics*.

3. INTERCONNECTED ESTHETICAL COLLECTIVITIES

The interconnections made of N nodes and L links model very well, in the sense given by Wittgenstein to the perception of structural self-organization, a collectivity. The nodes are the members of the collectivity that are tied by links. This type of collectivities we shall name, further, *interconnected collectivities*. The interconnected collectivities will not limit at the sets with the same type of nodes (resulting collectivities with non homogenous nodes) and/or at the sets with the same type of links (resulting collectivities with non homogenous links). What is certain, the structural entities, which form the collectivity, are *interconnected one way or another*. We should limit, without losing too much of generality, to the *orthogonal interconnections* or *orthogonal collectivities*. Any number of nodes of an interconnection can be represented as a product of integer numbers, $N=m_r \cdot m_{r-1} \cdot ... m_1$. On the basis of this representation, to each node of an interconnection we can associate an address X with r digits, $0 \leq X \leq N$-1. Further, we present some orthogonal interconnections as collectivities, i.e. sets selected or built by relations.

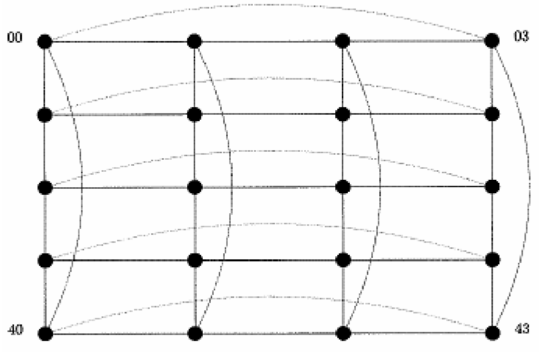


Figure 1: GHT Interconnected Collectivity

A *generalized hypercube*, GHC, is an orthogonal collectivity with $N=m_r \cdot m_{r-1} \cdot ... m_1$ nodes interconnected in r dimensions. In every dimension i of a collectivity the m_i nodes are interconnected all by all. The relation which establishes the interconnection of N nodes all by all is: the nodes addressed by $X = (x_r\, x_{r-1} \,...\, x_{i+1}\, x_i\, x_{i-1} \,...\, x_1)$ are connected addressed by $X' = (x_r\, x_{r-1} \,...\, x_{i+1}\, x'_i\, x_{i-1} \,...\, x_1)$, where $1 \leq i \leq r$, $0 \leq x'_i \leq m_i - 1$ and $x'_i \neq x_i$. The *hypercube*, HC, is a GHC with $N = m^r$. The *binary hypercube*, BHC, is a HC with $N = 2^r$ nodes, and the *completely connected structure*}, CCS, is another HC with $N = m$ nodes.

A *generalized hypertorus*, GHT, is another orthogonal collectivity with $N=m_r \cdot m_{r-1} \cdot ... m_1$ nodes interconnected in r dimensions. In every dimension i, $1 \leq i \leq r$, the m_i nodes being "collectivized" in a torus. The relation which establishes the r tori of GHT collectivity is: the nodes addressed by $X = (x_r\, x_{r-1} \,...\, x_{i+1}\, x_i\, x_{i-1} \,...\, x_1)$ are connected with the nearest neighbor nodes addressed by $X' = (x_r\, x_{r-1} \,...\, x_{i+1}\, x'_i\, x_{i-1} \,...\, x_1)$, where $1 \leq i \leq r$, $x'_i = | x_i \pm 1|_{\text{modulo } m_i}$. The *hypertorus*, HT, is a GHT with $N = m^r$ nodes and the *torus*, T, is a HT with $N = m$ nodes. BHC can be and HT with $N = 2^r$ nodes.

A *generalized hypergrid*, GHG, is, also, an orthogonal collectivity having $N=m_r \cdot m_{r-1} \cdot ... m_1$ nodes interconnected in r dimensions. In every dimension the m_i nodes are being collectivized in a *chain*, or, better said, every node X is connected in a *grid* with the nodes addressed by $X' = (x_r\, x_{r-1} \,...\, x_{i+1}\, x'_i\, x_{i-1} \,...\, x_1)$, $x'_i = x_i \pm 1|\ x_i \neq 0$ and $x_i \neq m_i - 1$; $x'_i = x_i + 1|\ x_i = 0$; $x'_i = x_i - 1|\ x_i = m_i - 1$, for $1 \leq i \leq r$. The *hypergrid*, HG, is a GHG with $N = m^r$ nodes. The *chain*, C, is a HG with $N=m$. A binary hypercube can be, also, a hypergrid with $N = 2^r$ nodes.

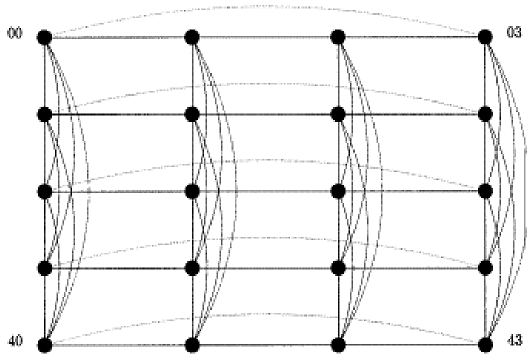


Figure 2: GHS Interconnected Collectivity

GHC, GHT and GHG are collectivities represented as *homogenous at links interconnections* or *homogenous interconnections* (the collectivities are homogenous at nodes, also; this paper does not refer to the non homogeneity at nodes). Most generally, the *non homogenous collectivities* can represent as non homogenous (at links) interconnections. Examples of non homogenous collectivities are the collectivities represented by *generalized hyper structures*, GHS, (Lupu 2004). A GHS is an orthogonal collectivity with $N=m_r \cdot m_{r-1} \cdot ... m_1$ nodes interconnected in r dimensions and in which every node X is collectivized (connected) in every dimension i, $1 \leq i \leq r$, to the nodes addressed by a *collectivizing (interconnecting) vector* $\left(\bigcup_{j=1}^{k_i} X^{ij}\right) = (x_r\, x_{r-1} \,...\, x_{i+1}\, x'_i\, x_{i-1} \,...\, x_1)$. $\left(\bigcup_{j=1}^{k_i} X^{ij}\right)$

specifies that a node of GHS is connected (non homogenous) by a *vector of elementary collectivizing structures* instead of a *single* structure in the homogeneous collectivities. This is non homogeneity at links of GHS specified by the collectivizing vector having, on the one hand, r elements, and on the other hand, k_i, $1 \le i \le r$, elementary collectivizing structures (homogenous) for which are specified the unions $\left(\bigcup_{j=1}^{k_i} X^{ij}\right)$, $j = 1, 2, \ldots, k_i$. So, X^{ij} are homogeneous elementary structures, like tori, grids, and chains, and must not be disjoint for a dimension.

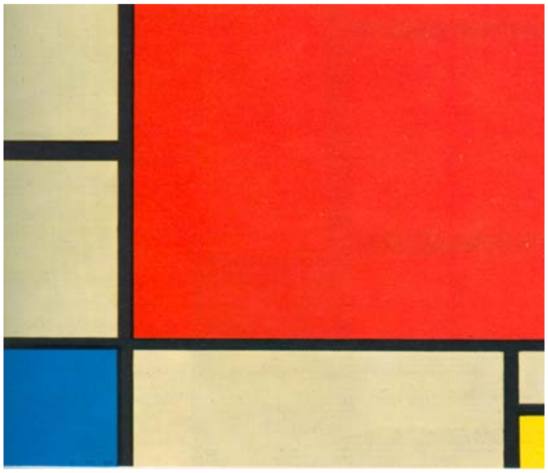


Figure 3: Mondrian, *Composition in red, blue and yellow*

In the figures 1 and 2 we give two examples of simple associations in collectivity modeled by a homogenous interconnection (fig. 1) and by a non-homogenous interconnection (fig. 2). At homogenous regular interconnections, as the GHC or HT, the origin position, “point of view”, does not matter. The collectivities, which they model, are *spherical*. The diameter is the same, doesn’t matter the point of view. At irregular networks, as GHG and other non-homogenous interconnections (e.g. GHS), it matters where the position of the origin is, it matters the point of view. The “structural” behavior around the origin at the collectivities modeled by these interconnections, is not spherical anymore. Why does the origin position matter? Because the structural non-homogeneity of an association in a collectivity from an origin is equivalent to a “functional potential” or, in this article’s case, an “esthetical potential” from the same point of view. For example, the more numerous and more varied the links in an interconnected collectivity from a point of view (a origin) are, the more sophisticated, more *adaptable* at a demand, or more *self-organized* the functions are. The interconnected collectivities, homogenous and non-homogenous, can be appreciated, at the beginning, by two general measures: the *locality* and the *globality*. The present paper refers only to the locality.

In the figure 3 we present an interconnected collectivity from the artificial esthetical world, an *esthetical interconnected collectivity*. It is a work from 1930 of Piet Mondrian, one of the first abstractionist painters. In the beginning Mondrian knew a cubist period, working in Paris with Braque and Picasso. It wasn’t long till he separated from them, because of his need to draw of cubism the “logical conclusions”, which they did not draw. Regarding the object, which is still visible in cubism, it could keep *the lines, the rhythm and the colors*, and order the painting canvas with only one aim, the *creation of an autonomous composition* (Muller and Elgar 1972). The Mondrian work (fig. 3), except the colors, may resemble with an orthogonal collectivity the nodes of which, in a first phase of study, are at the intersection of the colors. In the figure 4 we present the bidimensional interconnection that corresponds with the Mondrian composition from the previous figure.

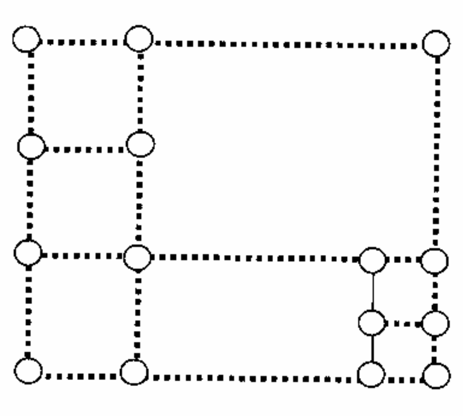


Figure 4: Interconnected Orthogonal Collectivity Overlapping to Mondrian Composition

4. ESTHETICAL LOCALITY

The collectivities structurally modeled by the interconnections (nodes and links) may be structurally estimated at the beginning, as primordial measures, by *locality* and *globality*. The *locality*, as we explained before, is *the spatial behavior of a collectivity around an origin*. As in Physics, where the gravity characterizes the attraction between objects, *the locality defines a collectivity*: the nearest the entities that compose the collectivity are, the best communicated, the best interfered, or in the case of the interconnected collectivities, the nearest the nodes are, the bigger the interconnection power is. In the esthetical collectivities, a bigger interconnection power can mean a bigger *expression power*. Consequently, the intuition of the structural self-organization of a work of art is bigger. Consequently, the intuition of a work of art is more intense. We name this kind of locality, *esthetical locality*. The esthetical locality helps us to understand (partially) an esthetical collectivity.

As we have explained in the introduction, the locality definition refers to the first sense of the structural concept, the connection between entities or, in interconnected collectivities (and esthetical ones), the links between nodes. Analytically, the locality in an interconnection measures by neighborhoods, neighborhood's reserve, *Moore* reserves and, synthetically, by diameter, degree or average distances. As any property which organizes the entities, the locality may be studied first *structurally* (topologically) and then *functionally*. In the present case, the esthetical functionality is replaced by the *expression*, as we have already explained. Therefore, the locality of an esthetical interconnected collectivity will be defined by two partial localities: a structural locality and an *expressive localit*y (which replaces the functional locality from my earlier works). The structural localities appreciate by the simplest measure: *neighborhoods*. The neighborhoods divide in *surface* (or radial) *neighborhoods* and *volume* (or spherical) *neighborhoods*. The surface neighborhood of an interconnected collectivity represents the entities, components or nodes number at the logical distance d, $SN_d(O)=N_d(O)$, where O is an arbitrary chosen origin. The volume neighborhood is $VN_d(O)= \sum_{i=1}^{d} N_d(O)$. The neighborhoods are analytical measures of the structural locality of an interconnected collectivity. But the structural locality can also be measured by synthetic measures, e.g. by diameter: at the same number of interconnected entities, the less the diameter is, the bigger the locality (in the meaning of the agglomeration) is.

The neighborhoods and the diameters are functions on the original position. At the collectivities interconnected in homogenous and regular structures, as the generalized hypercubes or hypertori are, the origin position does not matter. At the collectivities interconnected in irregular structures, as the generalized hypergrids and other non-homogenous structures are, it does matter where the position of the origin is. The topographic model presented in some of my previous works helped us to describe and, therefore, to study the "structural" behavior of the interconnected collectivities in homogenous and, especially, non-homogenous structures. The properties of the locality can be better "read" by the diameter contour patterns in the structural relief of an interconnected collectivity.

Besides the contour patterns, we have also introduced a measure which helps us to estimate this structural relief from the locality point of view: the *state of agglomeration*. The structural localities of an interconnected collectivity are more or less *agglomerated* and can be read by the help of the diameter contour patterns, as we have explained in the previous paragraph. The depth of the *valley* (minimum diameter) informs us about the *maximum agglomerated locality*, and the height of the *peak* (maximum diameter) about the *minimum agglomerated locality*. Thus, the *structural state of agglomeration of a node (entity) of an interconnected collectivity is given by the interconnection diameter computed with the origin in the corresponding node*. The contour patterns of the structural states of agglomeration constitute a map with the *structural relief of the interconnected collectivity*.

The structural locality is invariable information depending only on the topology of the interconnected collectivity. A point of view explicitly *expressive* on the esthetical locality can consider a parameter E_O, where O is the origin of the collectivity, which can depend on logical and physical distances between the collectivity entities (d_l, d_f), the colors of the entity (c), the movements of the entity (m), the "pictorial" message distribution (φ) or/and other factors.

Expressive locality of an esthetical collectivity is measured, as the structural locality, by neighborhoods: an *expressive surface neighborhood*, $ESN_d(O)= E_O\cdot N_d(O)$, and an *expressive volume neighborhood*, $EVN_d(O) = \sum_{i=1}^{d} E_O(d)\cdot N_i(O)$. The neighborhoods measure analytically the expressive locality. As for the diameter, in the case of the structural locality, there is a synthetic measure for the expressive locality of an esthetical collectivity, the *expressive average distance*. Through this average distance, we can give a definition to the *expressive state of agglomeration*: *the expressive agglomeration state of a node (entity) of an esthetical interconnected collectivity is given by the expressive average distance of the esthetical interconnection computed with the origin in the corresponding node*. The expressive agglomeration state is so much bigger as the expressive average distance is less. By the aid of the contour patterns of expressive states of agglomeration can draw a map, which depict the *expressive relief of the esthetical interconnected collectivity*. We shall refer to other works on the expressive locality.

The surface and volume neighborhoods, on the one hand, and the diameter or the degree, on the other hand, are analytical and synthetic evaluation means of the interaction capacity of an interconnected collectivity, measuring the *structural locality*. By the expressive neighborhoods and, synthetically, by the expressive average distance express which part of the structural locality is used in the esthetical process implemented on an esthetical collectivity. The expressive neighborhoods and the expressive average distances express the *expressive locality* of the esthetical collectivities.

To evaluate the structural locality of an interconnected collectivity, esthetical or not, near neighborhoods, we propose a simple and absolute measure of evaluation: *Moore reserve* based on *Moore* bound. As it is known, *Moore bound* is the maxim number of nodes which can be present in a *graph* given the degree l and the diameter D: $N_{Moore}=1+l\cdot(((l-1)^D-1)/(l-2))$. This bound is deduced from a complete l-tree having D diameter and is the *absolute limit* for the *diametric volume neighborhood*, $VN_D\ (O)=\sum_{i=1}^{D} N_d(O)$, in any graph (interconnected collectivity) with the degree l and the diameter D. Except for the complete l-ary trees, this bound is difficult to attain. *Petersen* graph, completely connected structures or the rings with odd nodes number, are interconnections reaching the *Moore* bound. Therefore, it makes sense to compute for an interconnected collectivity how far is this bound: the farther away the *Moore* bound is, the worse the structural locality is. This is implemented by the *Moore reserves*.

The *surface Moore reserve* is characterized by the difference between the number of nodes in a corresponding *Moore* tree at the distance d, with the degree l, and the surface neighborhood in the considered interconnected collectivity: $SMR_d=l(l-1)^{d-1}-N_d$. The *Moore reserve* is defined by the difference between the *Moore* bound at the distance d and the volume neighborhood: $MR_d=N_{Moore}(d)-VN_d$.

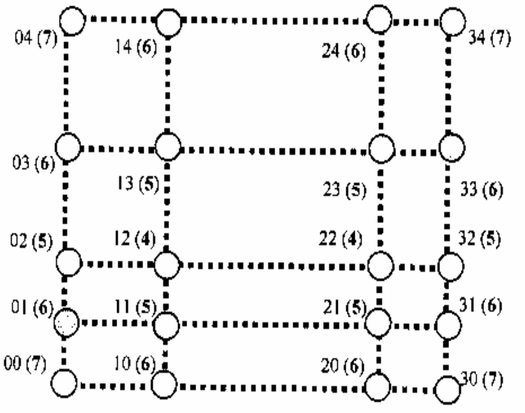


Figure 5: GHG Interconnection Corresponding to Figure 4

Let us come back at the bidimensional esthetical collectivity of fig. 4 and let us address the nodes corresponding to a mixed radix number system. From figure 5 results a "logical" GHG interconnection (logical because it does not take into consideration physical distances). GHG of fig. 5 is an interconnected collectivity with $N=m_1\times m_2=4\times5$ nodes, from which 5 are intersection points (nodes) "false", "non visible". The network is a kind of "logical" raster of Mondrian work specifying the visible and non visible "nodes" (the intersection points of the colors). The generalized hypergrid, GHG, is a non homogenous (non spherical) network, the structure of which is not the same, regarding each node as an origin. In brackets are written with bolds the diameters depending on the origin position or on the "point of view". The structural relief is like a valley or, better said, a *doline* in a karst areas and it is drawn in the figure 6. The maximum agglomeration (the bottom of the doline having the minimum diameter) is in the middle of the "logical" network where there are the two nodes with diameter 4. We notice that the two nodes are not invisible. Coming next, raising up towards the doline edge, there are six nodes (from which two are false) having the diameter 5, eight nodes (from which three are false) having diameter 6 and, finally, the corners of the network with diameter 7.

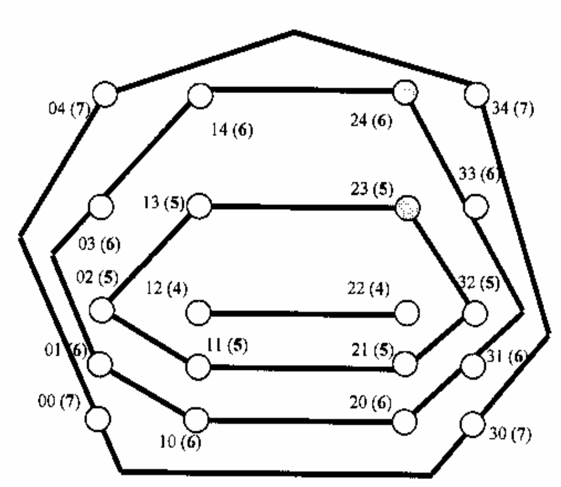


Figure 6: Structural Relief of Mondrian Painting Modeled with an Interconnected Collectivity GHG

Let us comment this distribution of states of agglomeration on the GHG collectivity corresponding to the Mondrian work. The maximum agglomeration (the minimum diameter, 4), an inverse "ridge" with two visible nodes (intersections of colors), is placed between the two of the most interconnected areas, on the left side and on the right side of the painting, in the "logical" middle of the interconnected collectivity. Climbing up to the doline edges, we come across a contour pattern with diameter 5 that have the invisible nodes *asymmetrically* arranged (an invisible node in the left colors intersections and an invisible node in the right colors intersections). The asymmetry of the invisible nodes increases at the contour pattern with diameter 6 towards the right-top side, the asymmetrical part of the painting. Mondrian leaves us, towards the right-top side, only with the painting edge, the red square, the biggest one. Mondrian painting is an asymmetrical work "as far as it is devoted to the worship of the *Imperfection*, deliberately leaving some things unfinished to complete by the play of the imagination" [Okakura, *Tea Book*]. In this way, the Asymmetry is a *structural communication*, a kind of a *structural dynamism* [Lupu, *Interconnecting*] in the *physical* collectivity representing Mondrian painting and in which there are two areas of *local* importance, the nodes {00, 02, 03, 04, 10, 12, 13, 14} and {20, 21, 22, 30, 31, 32}, placed asymmetricaly and non homogenously.

5. CONCLUSIONS

The (inter)connections are "patterns of discovery" (Alesso and Smith 2008). The interconnected collectivities are our models to esthetical behaviors. We have begun to model esthetical behavior by esthetical locality, a measure which can be estimated by neighborhoods, expressive states of agglomeration, expressive relief of the esthetical interconnected collectivity. We have exercised the esthetical model based on esthetical locality on an abstract painting of Mondrian. The esthetical locality makes the connection between the interconnection power and the expression power.

The preoccupations of artificial intelligence, artificial life and artificial sapience (Mayorga 2007) are the most well known. Why not an artificial esthetics?

ACKNOWLEDGMENTS

Many thanks to Prof. Tudor Niculiu, a good friend of mine and, at the same time, an erudite colleague, who presents this article. The author wish to thank the referees for their helpful comments and suggestions.

REFERENCES

Croce, B., 1971, *Estetics*, Bucureşti, Editura Univers.

Wittgenstein, L., 1991, *Tractatus Logico-Philosophicus*, Bucureşti, Editura Humanitas.

Castro, L. D. and Zuben F. V., 2005, *Recent Developments in Biologically Inspired Computing*, Brazil, Idea Group Publishing.

Drăgănescu, M., 1985, *Orthophysics*, Bucureşti, Editura Ştiinţifică şi Enciclopedică.

Nemoianu, V., 1967, *Structuralism*, Bucureşti, Editura pentru Literatură Universală,.

Okakura, K., 2007, *Tea Book*, Iaşi, Editura Dochia.

Lupu, C., 2004, *Interconnecting. Locality and Symmetry in Orthogonal Networks of Computers*, Bucureşti, Editura Tehnică.

Muller, J. E. and Elgar, F., 1972, *Un siècle de peinture moderne*, Paris, Fernand Hazan Èditeur.

Mayorga, R., Perlovsky, L. (Eds.), 2007, *Toward Artificial Sapience*, New York, Springer.

Alesso, H. P. and Smith, C. F., 2008, *Connections. Patterns of Discovery*, Hoboken, John Wiley & Sons, Inc.

BIOGRAPHY

CRISTIAN LUPU is Chief Scientist at the Romanian Academy, Centre for New Electronic Architectures, and Associate Professor at the Politehnica University of Bucharest, Faculty of Electronics. He received a M.S. degree in Electronics (Politehnica University of Bucharest, 1972) and a Ph.D. degree in Computer Science (Politehnica University of Bucharest, 1995). He has been with the Institute for Computer Technique in Bucharest (1972–1992) where he was Research and Development Engineer (1972–1979), Project Manager (1979–1989) and Head of Laboratory (1989–1992). He was among the first people who applied the technique of microprogramming in computer designing in Romania. Also, he was the first who designed a Romanian anti-skid system controlled by a microprocessor. As Professor of Computer Science he gives lectures about computer architecture and interconnection networks. In the last two decades, his main interest lies in the fundamental principles that underlay the evaluation/designing of the direct interconnecting networks, especially under structural point of view (Locality, Globality, Symmetry). He contributes in the area of interconnecting by emphasizing and studying Generalized Hyper Structures by a new topographic model. Dr Cristian Lupu is the initiator of some new theoretical concepts such as Group Locality, Interconnected Collectivities and Esthetical Collectivities. Dr Cristian Lupu has published more than 120 communications, articles, and reports. He has written seven technical books devoted especially to microprocessor designing and interconnection evaluation. For the book *Interconnecting. Locality and Symmetry in Orthogonal Networks of Computers* he received *Gheorghe Cartianu* Prize of Romanian Academy, in 2006. He has three patents in microprogramming and microprocessor applications. He is a member of ACM, IEEE/Computer Society/Technical Committee on Computer Architecture and IEEE SMC Society Industrial Committee. One may find the biographical profile of Dr Cristian Lupu in the *25th Silver Anniversary Edition* of Who's Who in the World/2008.

HIERARCHICAL INTELLIGENT ANALOG SIMULATION

Tudor Niculiu[(a)], Alexandru Copilău[(a)], Cristian Lupu[(b)], Anton Manolescu[(a)]

[(a)]University "Politehnica" of Bucureşti, Faculty of Electronics, Telecommunications, and IT
[(b)]Romanian Academy, Centre for New Electronic Architectures

[(a)] tudor-razvan@ieee.org, [(a)] c_alecs@yahoo.com, [(b)] cristianslupu@clicknet.ro, [(a)] tony@golana.pub.ro

ABSTRACT

Intelligence = Consciousness × Adaptability × Intention and *Faith* = Intuition × Inspiration × Imagination, are the complementary parts of the human mind. *Conscience* = Consciousness × Inspiration is the link between. Simulation is the relation between function and structure. Simulation of conscience demands transcending from computability to simulability by an intensive effort on extensive research to integrate essential mathematical and physical knowledge guided by philosophical goals. A way to begin is hierarchical simulation. Coexistent interdependent hierarchical types of different kinds structure the universe of models for complex systems. The symmetry between construction and understanding is an essential step to the symmetry between intuition and reason – extended adaptability for natural operations, and further, between faith and intelligence. Three examples confirm the assertions.

Keywords: abstraction, intelligence, consciousness, conscience

1. HIERARCHICAL APPROACH

Hierarchy types open the way to simulate intelligence as intentioned adaptable consciousness by extending the present limits of computability. We enrich the template concept to structures and create a theoretical kernel, for self-organizing systems, based on a hierarchical formalism. This permits theoretical development as well as efficient application to different cosimulation types of reconfigurable systems. Coexistent interdependent hierarchies structure the universe of models for complex systems, e.g., hard-soft ones. They belong to different hierarchy types, defined by simulation abstraction levels, modules, symbols, classes, and knowledge abstractions. Hierarchies of different types correspond to the kind of abstraction they reflect (↑):

- Class hierarchy (↑concepts) ↔ virtual framework to represent any kind of hierarchy
- Symbol hierarchy (↑mathematics) ↔ stepwise formalism for all (hierarchy) types
- Module hierarchy (↑managing) ↔ stepwise managing of all types by recursive decomposition, following the principle *Divide et Impera et Intellige*
- Construction hierarchy (↑simulation) ↔ simulation (design/ verification/ optimization/ integration) framework of autonomous levels for different abstraction grades of description
- Knowledge hierarchy (↑theories) ↔ reflexive abstraction aiming each level has knowledge of its inferior levels, including itself; this kind to abstract enables *consciousness*.

Knowledge and construction have correspondent hierarchy types: their syntax relies on classes, their meaning on symbols and their use/ action on modules. The hierarchy types can be formalized in the theory of categories (Niculiu and Lupu 2005). The hierarchical types are objects of equivalent categories (functorial isomorphic) that formally represent hierarchy types. The consciousness hierarchy type communicates to the other hierarchy types by countervariant functors, while the others intercommunicate by covariant ones.

Constructive type theory permits formal simulation by generating an object satisfying the specification. Applying similar abstraction kinds to hardware and software, representations and operations based on object-orientation, symbolization and structural abstraction can be extended from software to hardware.

A generic type - a form of polymorphism - is the ability to parameterize with types a hard/ soft element. Recurrence is confined to discrete worlds, while abstraction is not. This suggests searching for understanding following mathematical structures that order algebra into topology (Blum et al. 1998).

The alternative ways followed to extend the computability concept correspond to approaches known from German works: they respectively concentrate on the mental world of the good managed by technology, the physical world of the truth researched by science, and *Plato*'s ideal world of abstractions searched by arts.

1. Faust (*Johann Wolfgang von Goethe*): heuristics - risking competence for performance, basing on imagination, confined to the mental world.
2. Das Glasperlenspiel (*Hermann Hesse*): unlimited natural parallelism - remaining at countable physical suggestions, so in the Nature.

3. Der Zauberberg (*Thomas Mann*): hierarchical self-referential knowledge - needing to conciliate the discrete structure of hierarchy with the continuous reaction, hoping to open the way to Reality.

Note: | | signifies some sort of absolute difference.

$$\begin{aligned}
&\text{Simulation} \in \text{Behaviour} \times \text{Structure} \Leftarrow \text{Knowledge}\\
&\textit{Knowledge} \;\Leftarrow \textit{Intelligence :: information}\,()\\
&\textit{Imagination} \;\Leftarrow |\,\textit{Intuition - Consciousness}\,|\\
&\textit{Intention} \;\Leftarrow |\,\textit{Inspiration - Adaptability}\,|\\
&\textit{Adaptability} \;\Leftarrow \textit{simplifyingAbstraction}\,(\textit{Imagination})\\
&\textit{Consciousness} \;\Leftarrow \textit{reflexiveAbstraction (Intention)}
\end{aligned} \tag{1}$$

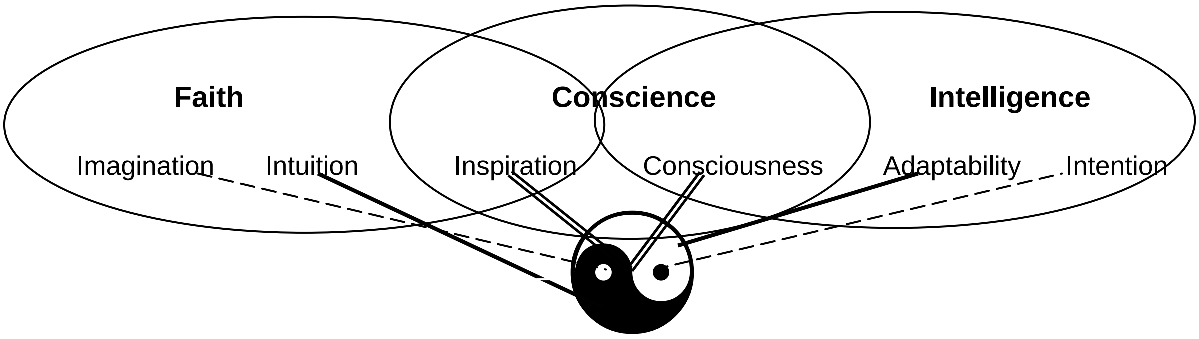


Figure 1: The Human Mind

2. LOOKING FOR CONSCIENCE

We follow the mathematical paradigm of intelligent simulation by functionally modelling the self-aware adaptable behaviour for intelligence simulation. The integration between discrete and analog is needed (Niculiu 2006), for a softer adaptability and for consciousness simulation as analog reaction. Recurrence of structures and operations enables approximate self-knowledge, with improved precision on the higher knowledge levels.

We oversimplify to move towards intelligent simulation: First, we neglect the essential but hard to understand intuition and inspiration, formalizing reflexive abstraction by knowledge hierarchy type and simplifying abstraction by construction hierarchy type:

$$\textit{Consciousness} = \textit{knowledge (simulation (Consciousness))} \tag{2}$$

The fixed-point relation suggests that we can model consciousness associating to any hierarchical level of the construction process a knowledge level. We have to build a metric space where knowledge ° construction is a contraction, i.e., elements implied in the construction get closer to one another in the formal understanding of the formal construct.

General functional relations between the essential parts of the faith-assisted intelligence imply:

$$\textit{Consciousness} = \textit{knowledge (intention (Inspiration, simulation (imagination (Intuition, Consciousness))))} \tag{3}$$

A continuous model for superior hierarchy levels offers a better model for consciousness of intelligence. Representation for design and verification is common ⇒ algebraic structures on which the different hierarchy types are based on are extended to topological structures; the different simulation entities are symbolic.

$$\textit{Reality is beyond Nature}\ (\text{IN} \subset \text{IR}) \tag{4}$$

The hierarchical principle is applied to the object of knowledge as to the knowledge structure itself: it mediates the action of a paradigm on an environment.

An intelligent system is capable of reflexive abstraction, controlled by problem specification and solving strategies. They are derived from a higher knowledge level, representing approach principles, structured by an even higher level containing abstract types.

Applied at environment and at simulation level, goal and strategies ensure flexibility of the framework, defining it precisely only in the neighbourhood of solved cases. For representation, the principle enables open modelling, which enables reconfigurable realization.

Formalizing hierarchical descriptions in continuous spaces we come closer to self-control, -organization, -awareness, i.e., (intention, adaptability, consciousness), hence to intelligence. There are enough positive signs for this from analog electronics, control systems, mechatronics.

Real progress towards this way of computation needs unrestricted mathematics, integrated physics, and thinking by analogies. Knowledge is based on morphisms between the real system and the simulator. An intelligent simulator learns generating and validating models.

$$\textit{Simulability is computability}^{\,continuum}. \tag{5}$$

Mathematics contains appropriate structures for self-referent models. The richest domain therefore is functional analysis, integrating algebra, topology and order, e.g., contractions and fixed points in metric spaces, reflexive normed vector spaces, inductive limits of locally convex spaces, self-adjoint operators of Hilbert spaces, invertible operators in Banach algebra.

Example: Let U be a universe that is structured by different hierarchies. U is a category, e.g., containing Hilbert spaces with almost everywhere-continuous functions as morphisms, enabling different ways to simulate self-organizing, -control, and -awareness. Hierarchical universe, functional objects (global functions, level structures, simplifying and knowledge abstractions), initial functions, and transformation rules define a hierarchical formal system. We consider the self-adjoint operators as objects on the higher levels of the knowledge hierarchy. These levels strive then for self-knowledge, whose degree rises as the knowledge abstraction, in the context of the inferior level knowledge, and of superior level qualitative knowledge. Natural transformations (functorial morphisms) on the functors of different hierarchy types solve the correspondence problem, i.e., the association of a knowledge hierarchy to the simulation one. Intention results by human-system dialog, and completes the simulation of the intelligence.

$$
\begin{aligned}
&(U, \{H_i \in S_h\}),\ card\ (U) > \aleph_0 && \text{// hierarchical universe}\\
&\Sigma = F \cup L \cup A \cup K && \text{// functional objects}\\
&\quad F = \{f \mid f \in U^* \to U\} && \text{// global functions}\\
&\quad L = \{f \mid f \in Level_j^* \to Level_j\} && \text{// level structures}\\
&\quad A = \{f \mid f \in Level_j^* \to Level_{j+1}\} && \text{//abstractions}\\
&\quad K = \{f | f \in Level_j^* \times Level_{j+1} \to Level_{j+1}\} && \text{// knowledge}\\
&I = \Sigma^* \cap R && \text{// initial functions}\\
&R = \{r | \ r \in \Sigma^* \times R^* \to \Sigma \times R\} && \text{// transformation rules.}
\end{aligned}
\tag{6}
$$

Further than modelling consciousness to simulate intelligence is the search to comprehend inspiration. A first idea is to use Lebesgue measure on differentiable manifolds and/or non-separable Hilbert spaces. Perhaps even mathematics will have to develop more philosophy-oriented to approach intuition.

God's ways are uncountable (7)

Evolution needs separation of faith and intelligence, understanding and using consciously more of faith's domain, and integrating them to human wisdom to be divided further to get more human.

His plans are hopefully hierarchical. (8)

3. ABSTRACTION AND HIERARCHY

The power to abstract is the crucial difference between human and other natural beings. *Divide et Impera et Intellige* applies the hierarchical expressed abstraction.

Intelligence and *faith*, like any dichotomy, can converge to integration or can destroy one another if not associated by *Conscience*.

Function is a transformation that can be mathematically formalized, or physically instantiated as temporal behaviour. *Structure* is a set of properties that characterize a mathematical or physical space. The properties can be constant/ variable in time - static/ dynamic structures.

Simulation is the relation between function and structure. Structured set = (Set, structure). *Model* results of an inversion-able representation of the simulation object. *Language/ system* is a generic form of a mathematical/ physical model. *Abstraction* is a human defining capacity that enables him to think.

The *simplifying abstraction* concentrates on a superior level the information that is considered essential for the current simulation approach. Reducing the informational complexity has in view to clear the operation and to ease its formalism; it can be only quantitative, but also qualitative.

The *reflexive abstraction*, expressed as knowledge hierarchy type, tries to understand itself better at higher levels, by understanding more of the inferior levels.

Hierarchy is a functional/ structural concept that fulfils mathematically/ physically the concept of abstraction. Hierarchy is syntax of abstraction.

God is in us - as faith is part of our definition, with us - by the others, and for us – the spiritual evolution, that is first conditioned, and then assisted, to be followed by the social one.

Against the danger of dichotomy, we concentrate in three different ways on the *unique Reality* (*Plato*): Art for the art - to look for the essential Way, Science with God's fear - to search for the existential Truth, and Engineering - to understand the Being and to concentrate more on the Spirit in our Life.

To go further, thinking while advancing, we divide twofold, as we cannot yet Intellige the dichotomies: *spirit-matter* (force-substrate, software-hardware) ⇒ real-natural (continuous-discrete, analog-digital), form-contents (category-functor, representation-simulation, class-function, structure-function), true-false, real-possible, perspective-profoundness, beauty-truth (arts-science, mathematics-physics) ⇐ *space-time* (evolution). Clearly, there should be no balance in most of the dichotomies. (Calude 2002)

Faith and Intelligence are ☯ *of Life.* (9)

Yin-Yang can represent by rotation any dichotomy. Arts and sciences are equally noble, even if one appears rather spiritual and the other rather material. Their alliance is vital and demonstrates the insolvability of the nowadays *spirit-matter* dichotomy, and of all resulted secondary dichotomies, actually functionally generated by the *space-time* dichotomy that is necessary to the human evolution. Reason is an extension of the nature. Nature is not an ephemeral context, but the matter we are built of in order to develop spiritually. The experiments for the spirit-matter dichotomy integration failed because of their extremism. The society is only the memory of the past, the manager of the present problems, and the assurance for a right future. We have to live together in respect of the others on the way to understand each other, in order to evolve toward essential beings for an integrated existence (see Appendix).

The present society is extremely materialistic, and tries to destroy every trace of ideal. We have to surpass the limits imposed by the essential dichotomy by a unique Ideal, named *God*, that should be constructive by continuous intelligent reconfiguration.

Human among humans should reflect a strategic equilibrium, without hiding or even violating, as happens nowadays, the principle that the society has to assist unconditioned the individual, with correct continuous education, and assistance by an intelligent faith to search and research the *unknown*.

The unknown can be interpreted as a *unique God*: the absolute freedom by understanding all the necessities, and the absolute unity by closing all the Divide et Impera et Intellige necessary for the Way to look for the Truth along the Life.

We extend the reconfigurability to the simulation itself (Figure 2). By a self-aware simulation, we get self-control of the simulation process. Therefore, we build a knowledge hierarchy corresponding to the simulation hierarchy. Then, by expressing both simulation and knowledge hierarchies in the reference system of the basic hierarchy types (classes, symbols, modules), we create the context for a self-organization of the simulation. The triad of the basic hierarchy types corresponds to the fundamental partition of the real life (beauty-arts, truth-science, good-engineering), that has to be continuously integrated by philosophy (essence, existence, being). The absolute functionality is symbolized by yin-yang, while the waves suggest hierarchical levels that are increasingly structured for simulation and knowledge.

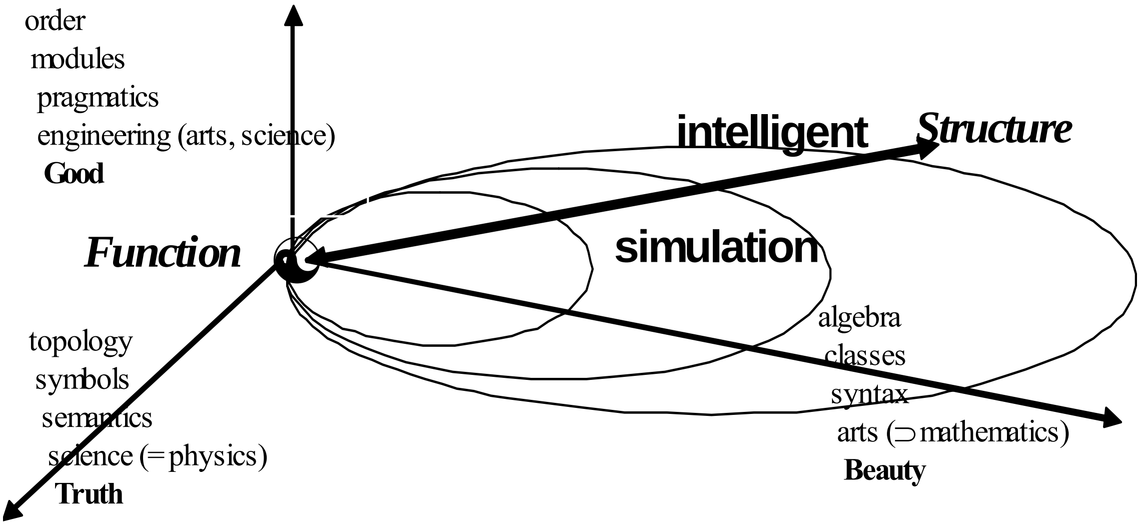


Figure 2: H-diagram
The basic hierarchy types (classes, symbols, modules) correspond to (syntax, semantics, pragmatics)
of the hierarchical language that has to express the intelligent simulation.
Intelligent simulation results from the integration of the simulation hierarchy
with its knowledge counterpart that represents a reflexive abstraction converging to self-consciousness
of the intended adaptable simulation.
The yin-yang represents the absolute functionality and the waves are increasingly structured hierarchy levels,
both for simulation as for knowledge.

4. INTELLIGENCE AND FAITH

Each of the nondeterministic separated complementary pairs is functionally structured like (interface, kernel, ambassador of the complement). The *yin-yang* model was not randomly selected: it is formed of three tangent circles emphasizing the centres of the inner ones. It retains only the essence of a dichotomy symbol suggesting a complete integration of the parts without loss of autonomy, realized by vicinity and pointing one to another. The Chinese symbol reflects the importance of something else, reminding of creation as love for something else. Three circles, each tangent to the others, models a partition of something to be understood in order to get further, says the centre of Europe. Circle is *cerc* only in our mother language, a perfect expression: *Cer* (sky) is the infinite, *cerc* is the finite representation of the infinite, by the permanent link from the (never)begin to the (never)end. π is the most famous real number (*Pythagoras*).

Cerc means perfection, which we permanently desire, therefore there exist integer numbers, having a perfect and beautiful theory, but not forgetting to continue the evolution searching and researching further. The western Europeans attain research/ rechercher by recursive search/ chercher. Our Romanian language helps us to approach this by *cercetare*.

The religion had to learn us about God's existence in our being. The philosophy has to learn us about essence, existence, and being. Our conscience is our representation of the essence of our existence as being, i.e., God is in ourselves, for ourselves, and among ourselves.

We have to be to search our essence (10)
researching our existence

Divide et Impera et Intellige has three parts as *alle guten Dinge sind drei* of the most philosophic European people. Mathematics develops by three basic structure types, integrating them. We divide our Universe in three worlds: essence, existence, and being. We divide our existence in three interdependent components: arts, science, and engineering-technologies, corresponding to our beauty-loving ideas, our truth-searching efforts, and our good-oriented constructions - presently exaggerated to exclusivity.

As the Reality contains abstract ideas, even if physics could explain everything as being discrete, the power of continuum cannot be forgotten. Consequently, the analog engineering has not to be neglected in modelling and simulation.

Physics permanently uses as dichotomy the discrete-continuous, while the engineering just adapts intuitively - as a primitive life form - to the requests of a consumption-oriented society - characteristic for primitive life. The reason is: either presently the engineering escaped the control of the inspiring arts as it also lost the consciousness that science conditions its existence, or that technology masters engineering as result of society being superior to human.

For physical or philosophical orientation, we need *cardinal* points. To inspire ourselves of the most pure of the arts, we learn about cardinal numbers (although, being sincere, mathematics leads the way to show that nothing is pure, so without leaving anything behind the Way has to be followed further). Cardinal numbers are just numbers of elements in a set, finite or not.

The Nature demands the least infinity and is defined by (0, successor, induction). Adding is in Nature's definition. However, the inverse operation, subtraction, needs negative numbers. We close mathematically the Nature to an Integer that opens the physics for recognizing the limits of Reason (electrons), in the meanwhile, attracting marvellous engineering solutions for different technologies. Electronics is among the most advanced engineering sciences; therefore, it has to be practiced by the most conscient human beings. Recurrent addition is multiplication, a most important parameter for the Nature. Mathematics closes the integers to the multiplication inverse, defining the rational numbers. These are not more than the naturals, but we can do many useful things with the Reason, from strategy to computer.

"What else do we need?" say too many, forgetting that the limits of the, so-called, pure Reason are caused by the fact that it bounds itself to close the Adaptability to (discrete) sequential operations. Thanks God, neither the mathematicians, nor the physicists do accept the all-happiness (Marcus 2000). They discover in three ways (order, algebra, analysis), which assisted all of them together to think, the power of continuum and that of the patience (Keutzer et al. 2000)

In this context, "mathematicians and physicists" means the theorem, natural laws, or even new approach discoverers, but also the engineers that understand the essential of mathematics and of physics.

We should not forget the third meaning of cardinal. It points to an unwise use of *Divide et Impera et Intellige* as a strategy called *when two fight, the third wins*. It means intervention only when the fighting forces begin to get unbalanced in favour of the less strong, not for establishing the equilibrium but for conquering both fighters. If the victory must be completed, both pseudo-ally and -enemy are firmly assisted, discretely or continuously, to loose control, because of all-happiness, respectively, all-unhappiness.

The 20th century is a too convincing example, unfortunately continuing to develop. Presently, we talk about electronic computers, but the nowadays trend is to copy from the living Nature, i.e., to emulate the living beings in unconsciously achieving complex duties.

Vanguard domains are biotechnology and computational intelligence. Neither intelligence nor life is well understood; remember *Zauberlehrling*. More important is that emulation is less human than simulation, so they should always develop in parallel, permanently exchanging experience.

The Reality does not reduce to Nature, as card (IN) is strictly inferior to card (IR). The Reason is the closure of the Nature relative to the primary operations, as $\mathbb{Q}$ is the closure of IN to the inverse operations of addition and multiplication.

However, the Reason is dense in Reality – as the reals are the analytical closure of the rationales, IR = $\{\lim_{n\to\infty}(q_n) | (q_n)\in \mathrm{IN}\to\mathbb{Q}\}$. The Reality extends beyond Nature and Reason, not just for the quality of the quantity, but also regarding the power of transforming operations. IR closes $\mathbb{Q}$ to the inverse of power rising – the last arithmetic operation resulted by recurrence of the prior one, which can be pursued by Reason. Further, closing to the inclusion order, the set of all subsets of IN, $\mathbb{Z}$, $\mathbb{Q}$ or in general, of countable sets, is the uncountable IR, the power of continuum.

To get to complex numbers is a matter of Imagination.

Reason closes the Nature to the inverse of natural
operations. Reality is the closure of the Reason
either to the inverse of artificial operations, (11)
or to the reasonably deducted infinite,
or even to an order over the Being itself.

We know that if there were no cardinal number between the natural/ integer/ rational discrete and that of the real continuum, then the logic would include the principle of the excluded tierce. This, pure and simple, hurts the Human, who is fond of nuances. Therefore, we can (*nonconstructively*) prove that there is an intermediary level between Reason and Reality. There are angels between Human and God said the wise. The density of Reason into Reality means that every real is the limit of a sequence of rationales.

Therefore, we hear nowadays that if we master the Reason, Reality becomes a complexity problem, i.e., speed of convergence. The density of ℚ in IR shows that between any two real there is a rational numbers one. Reality is much more than Reason can even imagine, but something reasonable exists between any two realities (*nonintuitive*).

Neither Intuition nor Reason arrives to something that nonconstructive mathematics proves elementary. As any true art or beautiful science of the ideas or the phenomena, mathematics does not limit itself to either Intuition or Reason, allowing them to collaborate by Conscience.

We dare use mathematics as metaphor for the relation between Nature and Reality, but it is only a correct inspiring analogy. IR is an initial step in mathematics for algebra, topology, order, and more, for their collaboration. Mathematics is for Reality just one of the favourite ways to get the Human closer to it.

The society is conservative – it tries to last forever at any evolution level, using a common measure. Everything can be evaluated, although most of the essential things on that our existence bases its being are not measurable.

The so-called pure Reason, i.e., the context-free Reason – most adaptable, conscious only for having, not for being, intended by the tactics of the consumption society, and totally unfaithful, gives the necessary force to stagnation or even to choosing a wrong way.

Unfaithful means here that the components of the Faith (inspiration, intuition, imagination) are used separately to serve the competition for the Good that makes present Life credible.

However, the irrational of arts, particularly in mathematics, is more than reasonable, whereby the society is less than reasonable; on the contrary, arts open the way to Reality by closure to an essential and radical operation. To master the New Power of the continuum is beyond Intuition and Reason, if they do not integrate by Conscience.

The adaptability-based Reason cannot explain or control thoughts, even if sequential is extended to unlimited parallel/ nondeterministic. These desired operational properties can be found mainly in the right side of the human mind.

Further, the difference between continuous and nondeterministic sequential is positive. Therefore, the Reason has to be Faith-dependent completed to Intelligence. A being needs more than Intuition and Adaptability to surpass the Matter by Spirit; only the integration of Intuition and Adaptability by Conscience can explain the Human being. All this inspire us to propose the thesis:

$$\textit{Conscience} = \textit{closure}_{(\textit{conscious simulation})}^{-1}(\textit{Conscience}) \quad (12)$$

Initially, Conscience = Consciousness

The idea can be formally sustained in the category theory (Ageron 2001). Informal arguments follow.

The essential limit of discrete computability, inherited by the computational intelligence, is generated by the necessity for self-reference to integrate the level knowledge with metalevel knowledge in Conscience modelling. A hierarchical type expressing reflexive abstraction can represent the conscient knowledge. The aspects of the Reality, and of the human mind reflecting it, are not to be neglected, although they are neither constructive nor intuitive. A way from Reason to Intelligence is to integrate Consciousness and Intention, then further Intelligence and Faith to become Reality-aware. We could consider just the simplifying types of hierarchy (classes, symbols, modules) and then express the construction, hoping to aim the absolute liberty, if we considered God as the simplest, unconstrained, essence of the Reality. However, we can simulate/ construct/ work/ live, associating consciousness hierarchies to all our activities, aiming to constructive understanding of the most complex absolute necessity, by this defining God. Abstraction is the human gift to go beyond natural limits, meanwhile extending pure reason to real intelligence → the metaphorical thesis:

God is the absolute abstraction (13)
→ the evolution goal for faith-assisted intelligence

5 PARTIAL CONCLUSIONS

1. *Hilbert spaces* ground the Behavioral model for quantum physics, i.e., the part that is independent of any concrete intervention (in the world of abstractions). The link to the complementary part of the model, representing the interface to the physical world, cannot be algorithmically expressed, what suggests that the model is not correct in the Reality.
2. *Banach algebra* introduce, additional to the topological vector spaces, a commutative multiplication; by an adequate transformation, a commutative functional composition results, eliminating an important sequential constraint.
3. *Simulability* and integrated mathemat/physical-comprehensive modelling the Intellige of the three approaches are promising ways.
4. *Inductive limits* direct the convergence of hierarchical types, enabling the compatibility of partial simulations and contributing to the correctness by construction of the design.
5. *Reflexive topological vector spaces* contain the necessary ingredients for the representation of the Conscience, by reflecting the adaptability in the variability of the space dimensions.
6. *Self-adjoint operators* and *eigenvalues/-vectors* assist the knowledge concentration/ stability.
7. *Fixed points* help to a formal simulation goal.
8. *Inseparable spaces* can instrument the understanding of inspiration and intuition.
9. *Simulability is computability to the power of continuum*: metaphorical thinking, unrestricted mathematics, e.g., mathematical measurability to formalize, and analog electronics.

Human = human (Humanity); // *class*
human $\in$ *Faith* $\times$ *Intelligence* $\rightarrow$ *Faith* $\times$ *Intelligence*
Humanity = ({humans}, eternal/ evolving Structure)
evolution $\in$ *(Hunger* $\times$ *Fear* $\times$ *Love)*
$\times$ *(Engineering* $\times$ *Science* $\times$ *Art)*
$\rightarrow$ *Engineering* $\times$ *Science* $\times$ *Art*
Mathematics $\subset$ *Art = Human:: beauty-oriented* (14)
activity (Science, Engineering)
Physics = natural $\cup$ *social Science*
= Human:: truth-oriented activity (Art, Engineering)
Engineering/ Engineering = Human::good-oriented
activity (Art, Science)

6 TRANSFER FUNCTION SINGULARITIES

Twenty years ago, one of the authors together with a friend - now professor at *Carnegie-Mellon* University - presented a related work that compared two methods to determine the poles and zeros of a transfer function, based on state-equations, respectively on node-equations (Marculescu and Niculiu 1987). Complexity of the set-up actions of the first was balanced by weak convergence of the second. This is a typical case to try heuristics together with expert systems.

Ten years ago, two of the authors together with the other members of a Romanian–German team, presented (Niculiu et al. 1997) a knowledge-based object-oriented analog simulation system.

The *Newton-Raphson* method was used in circuit simulation for forty years, and the interest for its optimization has not decreased. (Zu et al. 2007) The graphical or numerical results of a circuit simulator are the primary information that has to be sampled with a variable rate appropriate to the simulator output variation.

Knowing the dominant singularities is decisive for simulation, as they reflect the stability of the circuit (Manolescu 1999), or can represent primary information in formal simulation, e.g., root locus method.

The transfer function of a linear (linearly approximated around a static operation point) circuit is a ratio between real coefficient polynomials with complex roots, functionally describing the frequency behavior.

A pattern-matching search decides which rule applies, and at the end, the transfer function results as a two polynomials ratio. The search is bottom-up while determining the singularities, and top-down to find recursively the dominant ones.

The function of our program is threefold: *classification* - to recognize the type of singularity from the transfer function or *Nyquist* diagram; *control* – for stability; *anticipation* – to link the results to possible alternatives for improved behavior.

It is object-oriented, and written in Java. The main classes are *Element*, *Rule*, *Match*, and *Act*. The input is a circuit simulator .AC result (numerical or graphic), the output a rational function representing the approximate transfer function that describes the essential behavior.

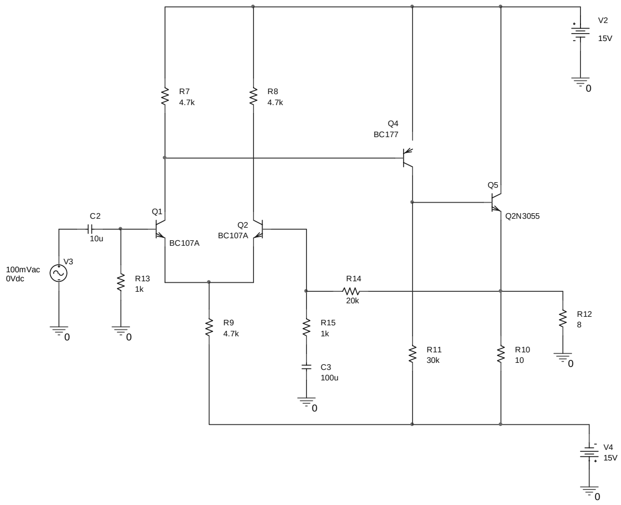


Figure 3: Audio Amplifier

For the integrated audio amplifier in Figure 3, the system finds the transfer function too “noisy”, and proposes to “clean” it, by insertion of RC series group in parallel to R_{14}; further it verifies whether the capacitance can be integrated.

Freedom is understood necessity (15)
Georg Friedrich Wilhelm Hegel

7 RECONFIGURATION

Representation is a 1-to-1 mapping from the universe of systems (objects of simulation) to a hierarchical universe of models; hence, a representation can be inverted. A model must permit knowledge and manipulation, so it has two complementary parts/ views: description and operation. In a formal approach models correspond to classes and specifications to instances.

Reconfiguration continues the ideas of hardware-software cosimulation, intending to extend the software flexibility to hardware, as parallel software tries to get closer to hardware performance.

The experimented ways to reconfigurable design are Field-Programmable Gate Arrays for circuits (Rabaey 1997) and reconfigurable networks for systems (Miller 1993).

Our project extends the reconfigurability to the simulation itself. Towards a self-aware simulation to control the simulation process we build a knowledge hierarchy corresponding to the simulation hierarchy, then by expressing both simulation and knowledge hierarchies in the reference system of the basic hierarchy types we create the context for a self-organization of the simulation (H-diagram). The basic hierarchy types correspond to essential views in languages/ systems theory, being derived from the main partition of our real life.

Reconfigurable computer architectures complement the existing alternatives of spatial custom hardware and temporal processors, combining increased performance and density over processors with application flexibility .

Recursive reconfiguration of the simulation process, at any hierarchy level, is allowed by different strategies that alter one of the technique/ model/ method if one of the imposed properties is not fulfilled after applying a technique, using a model and suitable methods for evaluation and reconfiguration. The process repeats for the initial description or the one resulted from prior (insufficient) improvement.

This calls for an intelligent control system that assists/ automates the reconfiguration. The techniques use hard-soft model templates, whose methods are recursively handling the different components in the system's description. Measurement functions control the continuation process of the reconfiguration, what suggested bringing reconfiguration in the context of software and hardware, as the strategies can be expressed object-oriented/ categorical and developed/ understood mathematically. Intelligent self-organization needs consciousness to control adaptability for reconfiguration.

We try to reach this goal integrating hierarchical intelligent simulation to nanotechnology realization.

```
class ReconfigurableSimulation { ...                          (16)
void reconf (Bool tech, Bool mod, Bool meth) {
    if (tech) {technique = selectTechnique (
                                        TechType techniques);
        if (mod) {model = technique.selectModel (
                                        ModelType models);
            if (meth) {method = model.selectMethod (
                                        MethType methods);
                (tech, mod, meth) = simulation (
                                technique, model, method);
                                                }}}} ...};
```

Simulation = (*representation*, *goal*) (17)

Dear God, search from the Sky
and see and research this Vineyard, (18)
implanted by Your Right, and complete it
(in eternity) Orthodox Pantocrator

CONCLUSIONS

1. The types of hierarchy link comprehension to construction: their syntax relies on classes, the meaning on symbols, and their use on modules.
2. The knowledge hierarchy type offers a way to model consciousness.
3. The theory of categories offers well-suited formalism for types.
4. Constructive type theory permits formal specification-verification generating an object that satisfies the specification.
5. The first idea is to consider/ remember that reality is infinitely more than nature.
6. Recurrence is confined to discrete worlds, while abstraction is not.
7. The difference suggests searching for understanding based on mathematical structures that order algebra into topology.
8. Especially, hierarchical reflexive ideas about ideas and how to get to ideas, representations on representations, objects to synthesize/ analyze/ modify objects, and how to build/ understand representations, concern the evolutionary intelligence.
9. We follow the paradigm of intelligent simulation functionally modeling self-aware adaptable behavior to simulate intelligence.
10. Our approach for singularities determination permits the most important aspect for the analog engineer: to know and to use the dominant singularities.
11. The integration between discrete and analog is needed, for a most soft adaptability and for conscience simulation as analog reaction.
12. Mathematics contains structures that suggest to be used for self-referent models.
13. The richest domain in this sense is functional analysis ← integrate (algebra, topology, order).

Neither intelligence nor life is well understood; remember *Goethe*'s *Zauberlehrling*. More important is that emulation is less human than simulation; remember *Mozart*'s *Zauberflöte*; they should always develop in parallel, permanently exchanging experience; remember *Thomas Mann*'s *Zauberberg*.

APPENDIX: THE PURE REASON EXPERIMENT

The Faith experiment, based on concentration, search, and construction, took place in the Middle Age by spiritual and chivalrous search, mediated by Masonic buildings. The Cathedrals were the symbol of the coming *revolutions* that intended to institute the constructive Faith as basis of the human society. The *USA* Constitution and *Napoleon*'s Code witness that the prepared superior level of the human-social evolution was not any sort of capitalism.

The society is conservative – it tries to last forever at any evolution level, using a common measure. Everything can be evaluated, although most of the essential things on that our existence bases its being are not measurable. The so-called pure Reason, i.e., the context-free Reason – most adaptable, conscious only for having, intended by the tactics of the consumption society, and totally unfaithful, gives the necessary force to stagnation or even to choosing a wrong way.

Unfaithful means here that the components of the Faith (Inspiration, Intuition, Imagination) are used separately to serve the competition for the Good that makes present Life credible. However, the society is less than reasonable, whereby, the irrational of arts, particularly in mathematics, is more than reasonable, opening the way to Reality by closure to an essential and radical operation. To master the *New Power* of the continuum is beyond Intuition and Reason, if they do not integrate by Conscience and do not collaborate by Imagination and Intention. The historical experiment of the pure Reason was the necessary intellectual condition of the first, and by now – the last, social revolution.

The initial goal of this event was a reintegration of the ways to search for the Spirit from the Matter (knights) and for the Matter from the Spirit (monks). It failed because it kept the arms, the wars, and the social classes, against it had risen.

More important, the experiment continued beyond its historical limits, what created the context to renounce to human dignity in order to reduce the human mind to adaptability and to throw Conscience and Faith into facultative.

The reduction of the constructive thinking to pure Reason weakened the human mind and made possible to restrict the point of views to the most dangerous of them.

The number of alternative paths, totally different but convergent to Reality, must be 3 – the last prime number successor of another prime. The concentration of the mind on the reasonable control of the Adaptability followed the spiritual revolution, which tried to bring into individual and social conscience that the human has chosen the evolution without disregarding the Eternity or knowing the Way.

The spiritual revolution selected a primitive form of *Divide et Impera et Intellige*, to begin researching what is partially known, leaving the unknown to be approached when the first step is finished. If this intention is not forgotten, the *Intellige* is contained in the *Impera* of the unknown that has to begin after the *Impera* of the partial known, with the completed knowledge that results.

This first step was done simultaneously by the institution that pretends to serve God - (*Luther*, the knight Popes), and by the most human Reality approach – the Arts (*Rinascimento*, *Descartes*). Their strategy was human-oriented. The contradictory sentence "to serve God" had sense as long as the Church tried to simulate the human conscience.

Perhaps was its partition thought as *Divide et Impera et Intellige* for the Way – Catholic, the Truth – Orthodox, and the Life – Evangelic, but there came no *Intellige*, and all of the alternatives fell into the exaggerating "*-ism*".

Perhaps this is analogous for Christians searching a beautiful Way, Jews researching a true Truth, and Buddhists engineering a good Life.

But many of us, of any religion, and respecting the traditions, are conscious of the Way to follow, do not expect anything from a metareal God (sounds like material), and are free to laugh even of their deepest Faith. Moreover, they are able to have a good Life, just enough to concentrate on the Truth and to follow a beautiful Way.

The concentration of the society on the material component of the human existence was necessary to liberate them of inhuman problems, not to attract the humans on secondary path. Antique Greece is an inspiring model (substituting slaves with intelligent systems). The Reason experiment had to finish 2 centuries ago, when:

- The pure Reason experiment climaxed by an unprecedented number of contemporary geniuses. This proved that people has to select wisely and to construct in good understanding and courageously a society that encourages/ assists them to evolve beyond the attained peaks: *Beethoven*, *Mozart*, *Gauß*, *Cauchy*, *Fourier*, *Laplace*, *Goethe*, *Schiller*, *Franklin*, *Kant* or *Hegel*;
- The cathedral builders tried to extend their work at a continental scale, neglecting the people on the building area, whose culture did not concentrate on *to have* but godly simple on *to be*;
- *Napoleon*, a genius of the military and social strategy art, showed that a new social form, reasonable in his plans, can not be imposed by the force against the revolution had fought.

We note that a century after Napoleon Bonaparte, a German genius of strategy, Otto von Bismarck, learning form his predecessors experience, was even more successful in unifying Europe. However, this time the materialistic forces were already masters of exploiting the instabilities, and hurried up to transform Europe in a laboratory to compromise any idealistic movement. They helped the generation of these movements and directed them to terrorism. As we said, the pure Reason experiment was of the form: complete the better known part (Bonaparte) to its limits (Bismarck), to have more chances beyond the limits. The falling and remaining in materialism hurt a lot both Nature and Human. The importance of the experiment was significant, but its continuation after the results could be interpreted has killed countless people and even cultures.

Nowadays the materialism torments increasingly, threatening the future. The adaptability-based Reason can not explain or control thoughts, even if sequential is extended to unlimited parallel/ nondeterministic (equivalent). Anyway, these desired operational properties can be found mainly in the right Faith-oriented side of the mind. Further, the difference between continuous and nondeterministic sequential (unlimited parallelism) is positive. Therefore, the Reason has to be Faith-dependent completed to Intelligence. A being needs more than Intuition and Adaptability to surpass the Matter by Spirit; only the integration of Intuition and Adaptability by Conscience can explain the Human being.

A way from Reason to Intelligence is to integrate Consciousness and Intention, then further to integrate Intelligence and Faith to become *Reality-aware*. Transforming the abstraction into comprehensive construction can be the *Goal* of the Human among Humans, *unique God* for different cultures of free humans. *Freedom is understood necessity* (*Hegel*). We have to remind our conscience to integrate our mind. We have to remind ourselves that society has the duty to assist humans to live among humans. We have to stop society to be more important the Human.

This is nowadays the case, and we are on the way to live in an aunt hill/ a swarm/ a herd/ a flock/ a stud, or even a pack/ a horde/ a crowd/ a mob. An operating system serves to autonomous programs both as link to the hard as for development of the soft. Analogous, the minimal unconditioned tasks of the society are health and education for everyone, encouragement for culture and researching for any Human - conscient human. The common measure was:

... ← ***Philosophy*** ← *human* ***Culture*** ← *special* ***Knowledge*** ← *economic* ***Force*** ← *physical* ***Force*** (19)

REFERENCES

Ageron, P., 2001. Limites inductives point par point dans les categories accessibles. *Theory and Applications of Categories* 8: 313-323.

Blum, L., Cucker, F., Shub, M., Smale, S., 1998. *Complexity and Real Computation*. NY: Springer.

Calude, C., 2002. *Information and Randomness*. NY: Springer.

Keutzer, K., Malik, S., Newton, A.R., Rabaey, J., Sangiovanni-Vincentelli, A., 2000. System-level design: orthogonalization of concerns and platform-based design, *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems* 19: 1523-1543.

Manolescu, A.M., 1999. *Analog Integrated Circuits*. Foton International: Bucureşti.

Marcus, S., 1999-2000. From Real Analysis to Discrete Mathematics and back: Symmetry, Convexity, Almost Periodicity and Strange Attractors. *Real Analysis Exchange* 25: 125-128.

Marculescu, R., Niculiu, T., 1987. Transfer Function Singularities Determination. *Proceedings of the IEEE International Conference on Semiconductors*, 61-64. October, Sinaia.

Miller, R., Stout, Q.F., 1993. Parallel Computations on Reconfigurable Meshes. *IEEE Transactions on Computers* 42: 321-340.

Niculiu, T. 2006. Evolution by Closure to the Inverse. *Noema Journal of Romanian Academy* V: 168-189.

Niculiu, T., Lupu, C., 2005. Categories for Hierarchical Simulation of Intelligence. *Proceedings of the IEEE EUROCON*, 233-236. September, Belgrad.

Niculiu, T., Manolescu, A.M., Manolescu, A., Glesner, M., 1997. Object-oriented Knowledge-based Hierarchical Analog Simulation. *Proceedings of the IEEE International Conference on Semiconductors*, 575-579. October, Sinaia.

Rabaey, J., 1997. Reconfigurable Computing. *Proceedings of the IEEE. International Conference on Acoustics, Speech and Signal Processing*, 127-132. October, München.

Zhu, Z., Peng, H., Cheng, C-K, Rouz, K., Borah, M., Kuh, E.S., 2007. Two-Stage Newton-Raphson Method for Transistor-Level Simulation. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems* 26: 881-895.

AUTHORS BIOGRAPHY

TUDOR NICULIU is Professor at the Electronics, Telecommunications, and Information Technology Faculty of the *Politehnica* University in Bucharest, and Senior Researcher at the Center for New Electronic Architectures of the Romanian Academy. He is looking for hierarchical integration of different domains, to understand intelligence by simulating it, and to apply it to intelligent simulation. Since 1991, he teaches and researches at the same institution, PhD 1995, MS 1985. Before, he was Senior Researcher at the R&D Institute for Electronic Components in Bucharest, researching and designing hierarchical simulation of analog integrated circuits. He studied Mathematics (University of Bucharest, MA 1994). He published 10 books and around 80 articles in international journals and conference proceedings. He is IEEE Senior Member of CAS, Computer, and SMC Societies.

ALEXANDRU COPILAU is Master Student at our faculty, studying Analog-Digital Hardware-Software Simulation.

CRISTIAN LUPU is Chief Scientist at the Romanian Academy, Center for New Electronic Architectures, and Associate Professor at the Faculty of Electronics, Telecommunications, and Information Technology of *Politehnica* University in Bucharest. He received 1972 MS degree in Electronics, 1995 PhD in Computer Science, from *Politehnica* University of Bucharest. He has been Head of Laboratory at the Institute for Computer Technology in Bucharest. His experience includes over 20 main projects in computer design, development tools, computer applications, computer simulation, microprocessor systems and applications, algorithms, and interconnecting networks. He published more than 120 articles, communications, and reports. He has three patents in microprogramming and microprocessor applications. He received 2006 the *Gheorghe Cartianu* Romanian Academy Prize, for the book *Interconnecting. Locality and Symmetry in Orthogonal Networks of Computers*. He is member of ACM, of IEEE Computer/ TC Computer Architecture, and of IEEE SMC/ IC.

ANTON MANOLESCU is Professor at the Electronics, Telecommunications and Information Technology Faculty of the University *Politehnica* in Bucharest. He researches, teaches and masters PhD students on Analog Integrated Circuits and Technologies. He published more than 100 articles in international journals and conference proceedings. He has three invention patents and a prize of the Romanian Academy for his outstanding activity. He is IEEE Member of Electron Devices Society.

SIMULATIONS WITH UNIFIED MOSFET MODEL FOR DISTORTIONS ANALYSIS

F. Babarada[a], A. Rusu[a], T. Niculiu[a], C. Ravariu[a], D. Vizireanu[a],
C. Moldovan[b], E. Manea[b], C. Dunare[c], M. Mlak[d]

[a] University "Politehnica" of Bucharest, Electronics Telecommunications and Information Technology, DCAE, ERG,
[b] National Institute for Research and Development in Microtechnologies, Bucharest, Romania,
[c] Department of Electronics and Electrical Engineering, University of Edinburgh, UK,
[d] Bucharest University of Economics, Romania.

[a]babflorin@yahoo.com, [b]carmen.moldovan@imt.ro, [c]camelia.dunare@ee.ed.ac.uk, [d]mlak.madalina@virgilio.it

ABSTRACT

The challenges of the integrated circuits and nanotechnology need very accurate models for active devices. From this point of view the design of linear analog circuits lacks models for state-of-the-art MOS transistors to accurately describe distortion effects. This is produced by the inaccurate modelling of the second order effects induced by high vertical gate field such as mobility degradation and short channel series resistance and second order effects induced by parallel drain field like velocity saturation in the ohmic region, channel length modulation in the saturation region, static feedback and weak avalanche. After a rigorous description of transistor transconductance and channel conductance in ohmic and saturation region we included these effects in the MOS transistor model, using a compact expression of drain current for computation reasons. The simulations using the new drain current expression were in good agreement with experimental data, proves scalability and large voltage range functionality.

Keywords: Modelling, Simulation, Design, Quantum mechanical effects.

1. INTRODUCTION

The advancement of knowledge in the electronic design is strongly influenced by Technology Computer Aided Design–TCAD. Here is an interesting positive feedback, because the computing power helps the designers to perform modelling, simulation, optimization and design of the new devices with improved performance which have the capability to increase the computing power.

The today integrated circuits arise the MOSFET like the most representative device. The continuum scaling, according with Moore's low, requires the permanent updating of the transistor physics (Magnus and Schoenmaker 2002) and for circuits models (Arora 1993), like the studies of current trough thin oxides and the modelling of insulated structures from substrate, named Silicon on Insulator-SOI. The devices models, used in circuits simulation programs, use frequently quasiempirical methods for optimisation precision/computing time report. Obviously these methods trend to be closely to the physical models (Fu and Willander 1999). So same parameters result directly from the physical structure or device technology like Fermi level, flat band voltage, mobility, length/width channel report, and so on. The simulation circuit programs push the physical models to the unified models, which can describe by one analytical expression the transistor function in any regime (Liou, Conde and Sanchez 1998). The continuous transition from weak inversion to strong inversion is realized by an appropriate interpolation function. The today MOS structures has the channel length in the submicron region but all the models begin with the long channel study and then make corrections about the channel length. That approach corresponds to the MOSFET scaling from the electrical behaviour point of view. The static characteristics will be maintained with the dimensions shrinking, but must be improved the dynamical characteristics. Other characteristic of MOSFET is the two dimensions functionality because the command by the gate is made transversal on the channel direction. In this case precise modelling can be made only with numerical methods but the numerical solutions can't be used for circuits analyse (Lundstrom 1990; Rusu 2001). The gradual approximation can describe separately the physical phenomena on both directions. This approximation describes with enough precision only the quasilinear regime of transistor, characterised by the continuity of the channel from the source to drain. In order to reduce the noise effects in analog and radio frequency integrated circuits the signal amplitude must be increased. The large amplitude will cause high distortion effects due to the inherent nonlinearity of the active devices used to implement the circuit functions (Ytterdal, Cheng, Fjeldly 2003; Babarada, Profirescu, Rusu and Dunare 2004). MOS transistors are the most adequate and effective devices for implementing analog functions. With the continuous downscaling of feature sizes the transistor characteristics are much affected by the second order effects such as mobility degradation and short cannel series resistance due to high vertical field, velocity saturation, channel length modulation, static feedback and weak avalanche due to the parallel electric field. All these effects will introduce additional nonlinearity in transistor characteristics and will finally produce output signal distortion. In the analog circuit design the MOS transistor can be driven as well by the drain terminal. By example a drain voltage driven MOSFET operating in the ohmic region like a gate voltage controlled resistor with the resistor linearity limited by the transistor drain voltage induced distortions.

Other example can be the load device in amplifier circuits where the drain terminal drives the MOSFET in saturation. The actual compact MOSFET models fail to predict satisfactorily drain induced distortion effects and the conductance g_d in the saturation region, which become important in amplifier circuits (Graaff and Klaassen 1990). The DC-bias conditions for which the third-order harmonic becomes zero give the minimum distortions. Contemporary MOS models do not predict these DC-bias conditions and an improved description of the drain current dependence of drain voltage is necessary. For analog design the characterisation include fitting of small-signal parameters such transconductance g_m and channel conductance g_d but this is not sufficient for proper estimation of distortion and intermodulation effects because for example the third-order harmonic distortion component depends on the third-order derivative of the device characteristics (Babarada, Profirescu, Ravariu, Manea, Dumbrăvescu, Dunare and Ulieru 2006). In the saturation region the gradual channel approximation is no longer valid, and as a result two-dimensional effects, such as channel length modulation and static feedback, come into play. In addition for large values of drain current, heat dissipation may become significant and thus self-heating also has to be taken into account in the drain current expression. Furthermore at high drain voltage values weak-avalanche effects become apparent and substrate current cannot longer be neglected.

2. SMOOTHING FUNCTION

Usual is assumed that drift current can be neglected in weak inversion and diffusion current can be neglected in strong inversion. Around threshold voltage is considered the moderate inversion region (Tsividis 1987), where both the drift current and diffusion current are important.

The transition from the ohmic region to the saturation region is continuous for the drain current. For analog circuit modelling, this transition should also be continuous for the higher-order derivatives of drain current to drain voltage (Babarada, Profirescu, Ravariu, Manea, Dumbrăvescu, Dunare and Ulieru 2006), or the model should be continuous. To arise these requires we replace drain-source voltage V_{DS} with an empirical function V_{DSsf} that changes smoothly from V_{DS} in the ohmic region to V_{DSsat} in the saturation region. This empirical function V_{DSsf} is usual named smoothing function. In order to preserve the model symmetry respectively the discontinuities of higher-order derivatives at V_{DS}=0V, we choose a smoothing function for which the derivate reported to V_{DS} is equal to unity, at V_{DS}=0V, (Brews 1978):

$$V_{DSsf} = \frac{V_{DS} \cdot V_{DSsat}}{\left(V_{DS}{}^{2m} + V_{DSsat}{}^{2m} \right)^{1/2m}} \qquad (1)$$

where: m is an empirical parameter, which can be integer only; V_{DS} is the drain-source voltage and V_{Dssat} is the drain-source saturation voltage.

The channel potential computation for the drain current is very difficult in moderate inversion and to ensure the continuity between the weak and strong inversion current and its derivatives an empirical approach is used. We used the next empirical interpolation function:

$$V_{GS} - V_T = 2 \cdot m_0 \cdot u_T \cdot \ln\left[1 + \exp\left(\frac{V_{GS} - V_T}{2 \cdot m_0 \cdot u_T} \right) \right] \qquad (2)$$

3. UNIFIED MODEL

The physics of solid begin from the characterization of all energy form using the Hamiltonian. After some approximations, like the unielectron assumption, we arrive at Maxwell's equations. Distinction between the drift and the diffusion component of the drain current should be maintained in all inversion regions, for an accurate description of the moderate inversion region:

$$I_D = I_{drift} + I_{dif} \qquad (3)$$

For model symmetry the drift current expression is given by usual formula (Babarada, Profirescu, Rusu and Dunare 2004), and making a Taylor expansion around $1/2\left(\psi_{sL} - \psi_{s0}\right)$, like in (Babarada, Profirescu, Ravariu, Manea, Dumbrăvescu, Dunare and Ulieru 2006):

$$I_{drift} = \beta\left(V_{GBeff} + \Delta V_G - \frac{\psi_{sL} - \psi_{s0}}{2} - \gamma\sqrt{\frac{\psi_{sL} + \psi_{s0}}{2}} \right)\left(\psi_{sL} - \psi_{s0}\right) \qquad (4)$$

where: β is the gain factor; $V_{GBeff}=V_{GS}+ V_{SB}- V_{FB}$, ΔV_G is determined by transition from drain induced barrier lowering-DIBL in weak inversion to static feedback in strong inversion; ψ_{sL} is the surface potential at the drain side; ψ_{s0} is the surface potential at the source side; γ is the body effect coefficient. The diffusion current is:

$$I_{dif}=\beta\cdot\gamma\cdot u_T\left[\sqrt{\psi_{s0}+u_T\exp\left(\frac{\psi_{s0}-V_{DS}-2\Phi_F}{u_T}\right)}-\sqrt{\psi_{s0}}\right.$$
$$\left.-\left(\sqrt{\psi_{sL}+u_T\exp\left(\frac{\psi_{sL}-V_{DS}-V_{SB}-2\Phi_F}{u_T}\right)}-\sqrt{\psi_{sL}}\right)\right] \tag{5}$$

and the total drain current expression, taking into account: drain saturation voltage, drain induced barrier lowering, mobility degradation, series resistance, velocity saturation, channel length modulation, static feedback and weak avalanche, is given by:

$$I_D=\frac{L\left(I_{drift}+I_{dift}\right)}{\left(\mu_0/\mu\right)\left(L_{eff}-\Delta L\right)+\left(\beta_R+\beta_{Th}\cdot V_{DS}\cdot V_{DSsf}\right)V_{GS-2}}\left(1+G_{av}\right) \tag{6}$$

where: L is the channel length; μ_o is low field bulk mobility; μ is carriers mobility (Babarada, Profirescu, Rusu and Dunare 2004); L_{eff} is the effective channel length; ΔL is the channel length modulation; β_R includes the series resistance; β_{Th} includes self heating; G_{av} is the weak avalanche, V_{GS-2} has the expressions:

$$V_{GS-2}=\frac{1}{2}V_{GS-1}+\frac{1}{2}\sqrt{V_{GS-1}^2+4\cdot\varepsilon^2} \tag{7}$$

where ε is a smoothing factor and V_{GS-1} has the next expressions:

$$V_{GS-1}=V_{GS}-V_{FB}-2\Phi_F-V_{DSsf}/2-\gamma\sqrt{2\Phi_F+V_{SB}+V_{DSsf}/2} \tag{8}$$

4. MEASUREMENT AND RESULTS

The higher-order derivatives have been performed by applying a sinusoidal signal to the terminal under investigation and by measuring the higher-order harmonics in the drain current, like in the configuration from fig. 1. The signal frequency is a few KHz in order to neglect the influence of capacitances. In this situation the distortions can be completely determined by the MOS transistor steady state.

The drain current I_D can be expanded in a Taylor series:

$$I_D=d_0+d_1\cdot v_i+d_2\cdot v_i^2+d_3\cdot v_i^3+\ldots=\sum_{i=0}^{\infty}d_i\cdot v_i^i \tag{9}$$

v_i is a sinusoidal signal $v_i=V\cdot\sin(\omega t)$ and the coefficients $d_i=\frac{1}{i!}\cdot\frac{\partial^i I_D}{\partial V_{G(D)S}^i}\bigg|_{V_{Do},V_{Go},V_{Bo}}$.

The drain current expression can be rewritten in terms of $\sin(n\omega t)$:

$$I_D=a_o+a_1\cdot\sin(\omega t)+a_2\cdot\cos(2\omega t)+a_3\cdot\sin(3\omega t)+\ldots \tag{10}$$

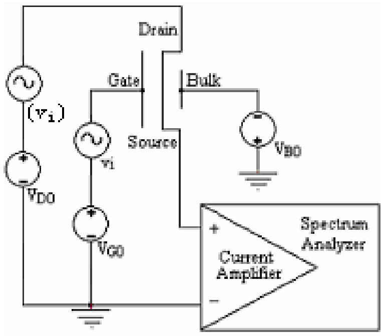


Figure 1: Distortion Measurement Configuration

The coefficients $|a_1|$, $|a_2|$ and $|a_3|$ are the signal harmonics amplitudes and were measured with the spectrum analyser, fig. 1. The coefficients can be written as:

$$\begin{aligned}a_1&=d_1\cdot V+\frac{3}{4}\cdot d_3\cdot V^3+\frac{5}{8}\cdot d_5\cdot V^5+\ldots\\a_2&=-\frac{1}{2}\cdot d_2\cdot V^2-\frac{1}{2}\cdot d_4\cdot V^4-\frac{15}{32}\cdot d_6\cdot V^6-\ldots\\a_3&=-\frac{1}{4}\cdot d_3\cdot V^3-\frac{5}{16}\cdot d_5\cdot V^5-\frac{21}{64}\cdot d_7\cdot V^7-\ldots\end{aligned} \tag{11}$$

For small enough values of the signal amplitude V, the coefficients (11) reduce to:

$$a_1\approx d_1\cdot V\ ;\quad a_2\approx-\frac{1}{2}\cdot d_2\cdot V^2\ ;\quad a_3\approx-\frac{1}{4}\cdot d_3\cdot V^3 \tag{12}$$

In this way the high order derivatives reported to V_{GS} notated g_{mi} or reported to V_{DS} notated g_{di} can be calculated or extracted from the measured harmonics.

Using the harmonic amplitude notation HD_i the second-order and third-order harmonic amplitude are:

$$HD_2=a_2\approx\frac{1}{2}\cdot d_2\cdot V^2=\frac{1}{4}\cdot\frac{\partial^2 I_D}{\partial V_{G(D)S}^2}\cdot V^2 \tag{13}$$

$$HD_3=a_3\approx\frac{1}{4}\cdot d_3\cdot V^2=\frac{1}{24}\cdot\frac{\partial^3 I_D}{\partial V_{G(D)S}^3}\cdot V^3 \tag{14}$$

Fig. 2-5 show the simulated and measured results for the third order derivatives $g_{m3}=\left(\partial^3 I_D\right)/\left(\partial V_{GS3}\right)$ and $g_{d3}=\left(\partial^3 I_D\right)/\left(\partial V_{DS3}\right)$, for n and p type MOS transistors, at V_{SB}=0V, for different device geometries W/L and W=10μm, like in fig. 6 and detailed in fig. 7.

A good fit between modelled and experimental results can be observed.

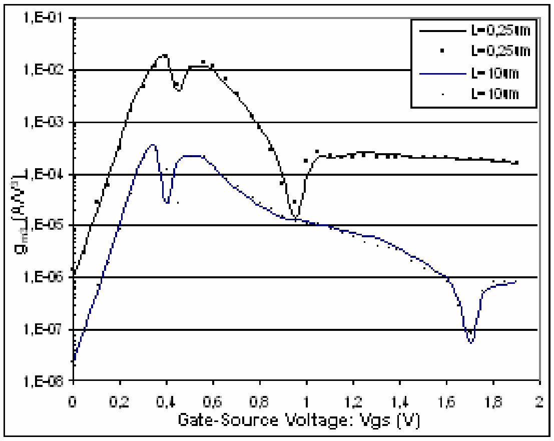


Figure 2: Modelled (Lines) and Measured (Symbols) Values of Gate Induced Distortion g_{m3} at Low Drain Bias (V_{DS}=50mV) as a Function of Gate Voltage for n-type MOS Transistor.

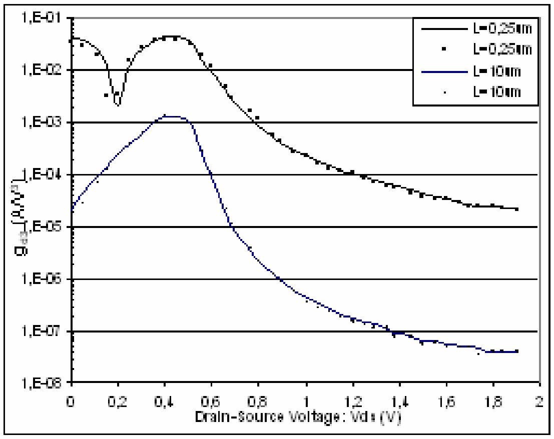


Figure 3: Modelled (Lines) and Measured (Symbols) Values of Drain Induced Distortion g_{d3} at V_{GS}=1V, as a Function of Drain Voltage for n-type MOS Transistor

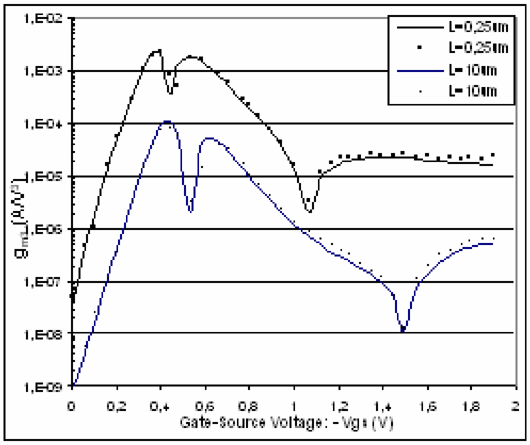


Figure 4: Modelled (Lines) and Measured (Symbols) Values of Gate Induced Distortion g_{m3} at Low Drain Bias (V_{DS}=-50mV) as a Function of Gate Voltage for p-type MOS Transistor.

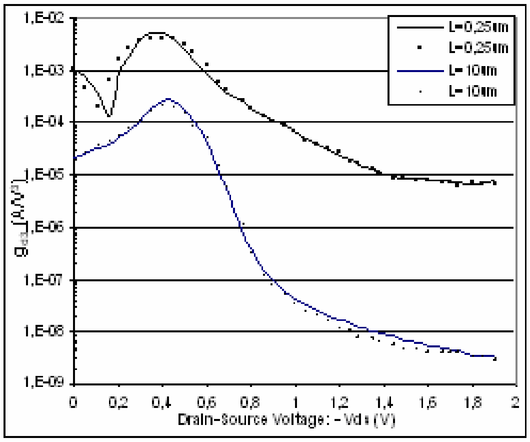


Figure 5: Modelled (Lines) and Measured (Symbols) Values of Drain Induced Distortion g_{d3} at V_{GS}=-1V, as a Function of Drain Voltage for p-type MOS Transistor.

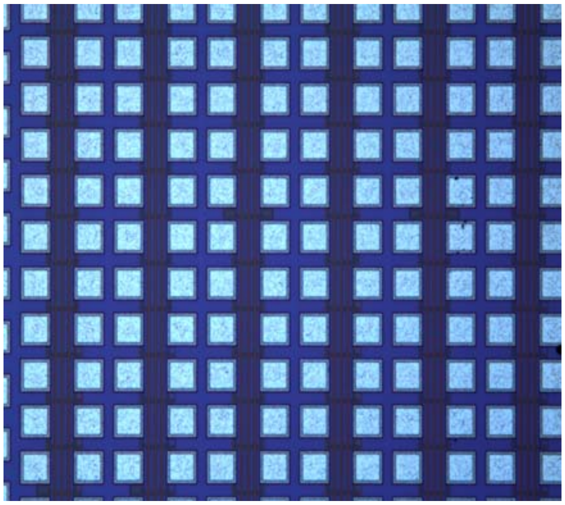


Figure 6: Different W/L Transistors Array.

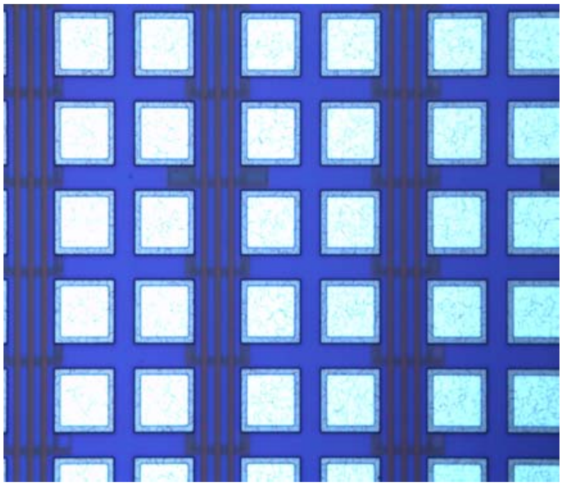


Figure 7: Detail of Different W/L Transistors Array.

The transistors array is performed in 0,25µm technology with transistors n and p enhanced type and

n^+ polysilicon gate for transistors type n and p^+ polysilicon gate for transistors type p. The gate oxide thickness is about 4nm and the maximum allowed supply voltage V_{DD} is 2,5V. The LDD structure is no longer used and description of gate voltage dependent series resistance become redundant and the model will be a constant source and drain resistance.

5. CONCLUSIONS

Precise physical description of channel surface potential is very important for precise modelling of drain current in all inversion regions. The unified model gives an accurate description of harmonic distortion in all inversion regions including drain saturation voltage, drain induced barrier lowering, mobility degradation, series resistance, velocity saturation, channel length modulation, static feedback and weak avalanche effects.

The model has a low number of parameters for both n and p type of transistors. The parameters extraction requires the measured current-voltage characteristics, which can be performed with usual testing tools.

The unified model is well scalable and the drain current and its higher order derivatives are precisely described over a large geometries range and terminal applied voltage values.

The model can be used for modern CMOS technologies in which both the n-channel and p-channel transistors are of the enhancement-type and LDD-structures are no used, it was found that an adjusted expression of hole mobility and a constant series resistance have to be used. In this manner the unified model can be used with good results even for deep submicron technologies.

REFERENCES

Arora, N., 1993. *MOSFET Models for VLSI Circuit Simulation, Theory and Practice*. Wien:Springer-Verlag.

Babarada, F., Profirescu, M., Rusu, A., Dunare, C., 2004. Carrier Mobility and Series Resistance MOSFET Modelling. *BioMEMS and Nanotechnology-SPIE*, vol. 5275-49, pp. 354-363. 2004, Perth Australia.

Babarada, F., Profirescu, M., Ravariu, C., Profirescu, O., Manea, E., Dumbrăvescu, N., Dunare, C., Ulieru, D., 2006. MOSFET Modelling Including Second Order Effects for Distortion Analysis. *Applied Simulation and Modelling-IASTED*, pp. 506-510. 2006, Rhodes Greece.

Brews, J., 1978. A Charge-Sheet Model of the MOSFET. *Solid-State Electron*, Vol. 21:pp. 345-355.

Fu, Y., Willander, M., 1999. *Physical Models of Semiconductor Quantum Devices*. Kluwer/ Academic Publishers.

Graaff, H.C., Klaassen, F.M., 1990. *Compact Transistor Modelling for Circuit Design*. Wien:Springer-Verlag.

Liou, J.J., Ortez-Conde, A., Garcia-Sanchez, F., 1998. *Analysis and Design of MOSFETs, Modeling Simulation and Parameter Extraction*. Kluwer.

Lundstrom, M., 1990. *Fundamentals of Carrier Transport*. Addison-Wesley.

Rusu, A., 2001. Toward a Demonstration of the Non-Linear Electrical Conduction Law. *International Semiconductor Conference*, pp. 421-424. 2001, Sinaia Romania.

Tsividis, Y.P., 1987. *Operation and Modeling of the MOS Transistor*, New York:McGraw-Hill.

Ytterdal, T., Cheng, Y., Fjeldly, T., 2003. *Device Modeling for Analog and RF CMOS Circuit Design*. England:Jhon Wiley.

Magnus, W., Schoenmaker, W., 2002. *Quantum Transport in Sub-Micron Devices*. Germany:Springer-Verlag.

AUTHORS BIOGRAPHY

Florin N. Babarada received the M.Sc. degree in Electronics & Telecommunications in 1983 and the Ph.D. degree with dissertation “Microresonant Sensors in Planar Silicon Technology” in 2000, both at University “Politehnica” from Bucharest, Romania. In 1984 he joined to Technology Department from Miroelectonica S.A. IC factory. From 1994 was member of the beginning team of the Microelectronics Institute-IMT and then named National Institute of Microtechnology, where he worked in the field of Microelectronics IC and Microelectromechanical Systems-MEMS. From 2002 he work at the University “Politehnica” of Bucharest, Faculty of Electronics Telecommunications and Information Technology, Department of Devices Circuits and Electronics Apparatus, where he’s involve in: Modelling, Simulation and Electronic Design Automation (EDA) of Micro/Nanoelectronics and Bio-Micro/Nanoelectronics Processes, Devices, ICs, MEMS and NEMS; Software methodology CAD, TCAD and Computer Graphics in Electronics and Bioelectronics Engineering; Computer aided engineering education e-Learning and e-Training. email: babflorin@yahoo.com

Adrian RUSU is the head of the Devices Components and Electronics Apparatus Department from University Politehnica of Bucharest, Faculty of Electronics Telecommunications and Information Technology, B-dul Iuliu Maniu 1-3, Sector 6, Bucharest Romania. His researching area concerns new electronic devices, technologies and non-linear conduction through semiconductors. email: adrianr@mcma.pub.ro

Tudor NICULIU is with the Politehnica University of Bucharest, Romania, Faculty of Electronics Telecommunications and Information Technology, Microelectronics Department, B-dul Iuliu Maniu 1-3, Sector 6, Bucharest. He is Senior IEEE Member with high performances in artificial intelligence, mathematical algorithms, simulation. e-mail: tniculiu@yahoo.com

Cristian RAVARIU is with the Politehnica University of Bucharest, Romania, Faculty of Electronics Telecommunications and Information Technology, Microelectronics Department, B-dul Iuliu Maniu 1-3, Sector 6, Bucharest, Postal zip: 061071. In the lasts years he studied nano-electronic devices, proposed a Nothing On Insulator configuration, besides to Bioelectronics devices, cellular nano-electronics. email: cravariu@arh.pub.ro

Dragos Vizireanu is with the Politehnica University of Bucharest, Romania, Faculty of Electronics Telecommunications and Information Technology, Telecommunications Department, B-dul Iuliu Maniu 1-3, Sector 6, Bucharest. He is Senior IEEE Member with high performances in digital signals processing, images processing and forms recognition, electric and electronics measurements.
e-mail: dvizireanu@yahoo.com

Carmen Moldovan graduated the Faculty Electronics and Telecommunication in 1983 and she owns a PhD in Microsensors. She worked to Microelectronica SA CMOS integrated circuits factory and now to National R&D Institute for Microtechnology, Bucharest, like Head of the Microtechnology Department. She is a member of: IEEE and Science and Technology Commission of the Romanian Academy and NEXUSPLUS and BRIDGE subcontractor (and also a member of the NEXUSPLUS Steering Committee). The scientific activity is published in more than 55 papers in journals, books and communications in Proceedings. email: carmen.moldovan@imt.ro

Elena Manea graduated the University of Bucharest, Faculty of Physics. Then she worked to Microelectronica, Bucharest, MOS I.C. Wafer Fabrication Division, like Head of MOS I.C. Wafer Fabrication Division. Now work to National R&D Institute for Microtechnology, Bucharest, Wafer Fabrication Laboratory, like Ph.D, senior scientific researcher and Head of Wafer Fabrication Laboratory. She has over 20 years experience in silicon planar technology including: photolithography, diffusion processes for MOS-CI, thin film deposition e.s.o. email: elena.manea@imt.ro

Camelia Dunare graduated the Faculty Electronics and Telecommunication in 1984. She worked to Microelectronica SA, National R&D Institute for Microtechnology and The Scottish Microelectronics Centre (SMC), University Edinburg GB. The research field is silicon processing, MOS and MEMS processes development. email: camelia.dunare@ee.ed.ac.uk

Madalina Mlak graduated the University Politehnica of Bucharest, Faculty Automatization and Computers in 1993. She is Assistant to Bucharest University of Economics, Romania, Informatics Department. She works in the field of computer network management of physics, logical and informational resources, network security, interactive interfaces, and numerical computing methods. email: mlak.madalina@virgilio.it

FAILURE PROCESS SIMULATION OF A COMPONENT-BASED SOFTWARE

Florentina Suter

University of Bucharest
Department of Mathematics and Computer Science
and Centre of Mathematical Statistics of the Romanian Academy

florentina.suter@g.unibuc.ro

ABSTRACT

In order to characterize as realistically as possible the evolution of software in time, the software reliability models should take into account the structure of the software. Such models are component-based models in which the software is not a black-box, but has several interconnected components. For this kind of software reliability models, due to their complexity, mathematical tractability becomes difficult to obtain. Therefore, thanks to its flexibility, simulation is a natural choice for analyzing the failure process and for estimating software reliability. Recently some software reliability models which take into consideration the modular structure of the software were described. In this paper we take as starting point a component based software reliability model, we describe a generalization of it and we use discrete-event simulation to analyze the software failure process.

Keywords: software reliability, component-based software model, discrete event simulation

1. INTRODUCTION

The main steps in the development of a software product are: requirements identification and analysis, project development, code writing and product testing. At the testing step the software is executed and at each failure one or more faults are discovered and corrected. It is not possible to correct all the faults from a software, and an important decision to be made in the testing phase is when to stop testing and release the software on the market.

An element to support this decision can be the software reliability, that is, the probability of failure free software functioning for a certain period of time, in certain environment conditions. For this reason many software reliability models were developed. These models are based mainly on two principles: the software product is a black-box and the software reliability is estimated from the probability assumptions on the software failure process. For example several software reliability models are based on the hypothesis that the counting process $\{N(t), t\geq 0\}$, where $N(t)$ is the number of the failures experienced in the time interval $(0,t]$, is a nonhomogeneous Poisson process. In (Chen and Singpurwalla 1997) a large number of software reliability models are unified by the observation that the failure times of the software are the jump times of a self-exciting process which is a generalization of the Poisson process and of the pure birth process.

An example of software reliability model based on a self-exciting point process of memory 1 can be found in (Al-Mutairi, Chen and Singpurwalla, 1998). More precisely the counting process $\{N(t), t\geq 0\}$ has the intensity functions:

$$\eta_0(t) = \frac{1}{(t/c)+(1/b)} \tag{1}$$

$$\eta_i(t;t_i) = \frac{1}{[(t-t_i)/c]+(t_i/ib)}, \quad i = 1,2,... \tag{2}$$

with b and c two positive parameters.

Recently a new approach of software reliability models is gaining importance. This approach takes into account the modular structure of a software. Some of the advantages of this approach are that it relates system reliability to its structure and the components reliabilities and it allows the analysis of the sensitivity of system reliability to the reliabilities of its components (Gokhale and Lyu 2005). The disadvantage is that, due to their complexity, mathematical tractability for this type of models becomes difficult to obtain. Therefore, thanks to its flexibility, simulation is a natural choice for analyzing the failure process and for estimating software reliability. Software reliability models which take into consideration the modular structure of the software are described in (Gokhale and Lyu 2005; Yacoub, Cukic and Ammar 2004). We take as starting point a model introduced in (Gokhale and Lyu 2005), we describe a generalization of it and we use discrete-event simulation to analyze the software failure process.

2. A COMPONENT-BASED SOFTWARE RELIABILITY MODEL

In (Gokhale and Lyu 2005) some component-based software reliability models are introduced in order to analyze the software failure process. We will focus on the model for which some assumptions from classical software reliability models are maintained considering that at each failure corresponds only a fault and that a fault is instantaneously corrected. There are also assumptions that take into account the structure of the software, thus the black-box hypothesis being relaxed. These assumptions are:

1. The software product has *k* components.
2. The components are executed sequentially beginning with component 1 and terminating with component *k*.
3. The component *j* is executed upon the completion of component *i* with the probability p_{ij} .
4. The failures of a component are independent from the failures of others components.
5. The time spent in each component is a random variable.

Moreover, one supposes that the behaviour of each component from the failure process point of view is modelled using a well-known software reliability model based on the nonhomogeneous Poisson process, the Goel-Okumoto model.

Taking into account all these assumptions it results a model which is complex and difficult to handle from computational point of view. Therefore in (Gokhale and Lyu 2005) the discrete-event simulation is used to study the failure process of a software whose structure and behaviour are described by the above hypothesis. As events they consider the transfer of control among components and the failure of the components. It is a rate-based simulation which has as input: the length of testing time duration, the time step and the failure rate of the component. The simulation procedure returns the total number of the failures observed in the time interval (0,*t*].

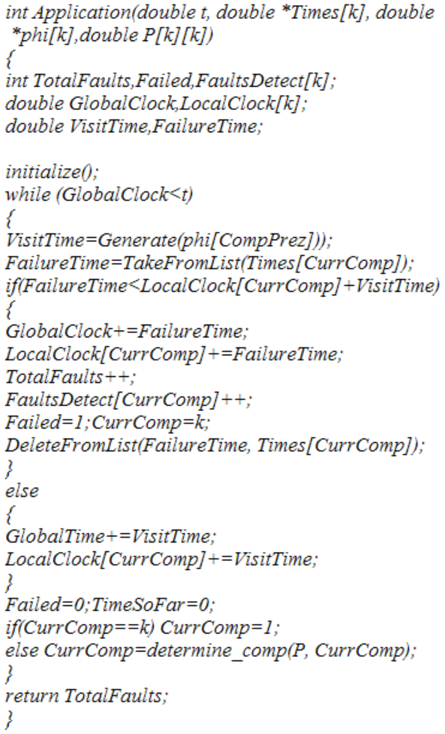
3. SIMULATION OF A SOFTWARE TESTING PROCEDURE

We consider the previous described model and we generalize it in the following way:

- instead of generating possible failures for some constant time intervals, we generate failure times using a generalization of inverse method;
- instead of considering the Goel-Okumoto model as a software reliability model for each component we consider a more general model in which the failure process is a self-exciting process.

In this way we avoid the possible problems that the choice of the length of time step could create. Moreover using the self-exciting point process we obtain a more general simulation model, such that we can associate to software components any software reliability model that is generalized by self-exciting process model. We analyze the application of our model in the case in which the failure process of the components has the characteristics described in (Al-Mutairi, Chen and Singpurwalla, 1998).

Our application is:



```
int Application(double t, double *Times[k], double
 *phi[k],double P[k][k])
{
int TotalFaults,Failed,FaultsDetect[k];
double GlobalClock,LocalClock[k];
double VisitTime,FailureTime;

initialize();
while (GlobalClock<t)
{
VisitTime=Generate(phi[CompPrez]));
FailureTime=TakeFromList(Times[CurrComp]);
if(FailureTime<LocalClock[CurrComp]+VisitTime)
{
GlobalClock+=FailureTime;
LocalClock[CurrComp]+=FailureTime;
TotalFaults++;
FaultsDetect[CurrComp]++;
Failed=1;CurrComp=k;
DeleteFromList(FailureTime, Times[CurrComp]);
}
else
{
GlobalTime+=VisitTime;
LocalClock[CurrComp]+=VisitTime;
}
Failed=0;TimeSoFar=0;
if(CurrComp==k) CurrComp=1;
else CurrComp=determine_comp(P, CurrComp);
}
return TotalFaults;
}
```

Figure 1: A software failure process simulation model

As in (Gokhale and Lyu 2005) our application simulates the execution of a software with *k* components and returns the total faults detected in the time interval [0,*t*). The input parameters for our application are

- *t* the length of the testing time interval;
- **Times[k]* an array in which each element is a list of failure times of a component
- **phi[k]* an array containing information regarding execution time spent in each component
- *P[k][k]* a matrix whose elements are intercomponent transition probabilities.

We suppose that the software execution begins with the component number 1. The testing time of each component is randomly generated. If an error occurs, i.e. there is a failure time whose value is in the testing time interval of that component, then the error is counted and the test restarts with the execution of the first component. If any error occurs, next component to be tested is chosen using the transition matrix *P[k][k]*.

The main difference between our model and the model introduced in (Gokhale and Lyu 2005) is that we generate failure times of different components. More precisely we generate random vectors whose elements represent failure times in time interval $[0,t)$ of one component of the software. For the generation of these random vectors we can use the generalized inverse method (Văduva, 1994). We can apply the generalized inverse method for generating jump times of a self-exciting point process for which the intensity function is known (Suter, 2004). For example if we apply the generalized inverse method to the point process considered in (Al-Mutairi, Chen and Singpurwalla, 1998) we obtain the following relationships for generating jump times of the point process:

$$\hat{T}_1 = \frac{c}{b}\left(U_1^{-\frac{1}{c}} - 1\right) \tag{3}$$

$$\hat{T}_i = \frac{c}{b}\hat{T}_{i-1}\left(U_i^{-\frac{1}{c}} - 1\right),\ i = 2,3,\dots,n \tag{4}$$

where U_1, U_2,…,U_n are independent uniform random variables.

REFERENCES

Al-Mutairi, D., Chen, Y., and Singpurwalla, N. D., 1998. An adaptive concatenated failure rate model for software reliability. *Journal of the American Statistical Association*, 93 (443), 1150–1163.

Chen, Y., and Singpurwalla, N. D., 1997. Unification of software reliability models by self-exciting point processes. *Advances in Applied Probability*, 29 (2), 337–352.

Gokhale, S., and Lyu, M., 2005. A simulation approach to structure-based software reliability analysis. *IEEE Transactions on Software Engineering*, 31 (8), 643–657.

Snyder, D. L., 1975. *Random point processes*. John Wiley & Sons.

Suter, F., 2004. *Models and Algorithms in Systems Reliability*. Thesis (PhD), University of Bucharest.

Văduva, I. (1994) *Fast algorithms for computer generation of random vectors used in reliability and applications*. Tech. Rep. 1603, Technische Hochschule Darmstadt, Zentrum für Praktische Mathematik.

Yacoub, S., Cukic, B., Ammar, H., 2004 A Scenario-Based Reliability Analysis Approach for Component-Base Software. *IEEE Transactions on Reliability*, 53 (4), 465–480.

EVALUATION OF GA ESTIMATION PARAMETERS IN WIDEBAND PIEZOELECTRIC TRANSDUCERS EMPLOYING A COMPREHENSIVE MODELING TOOL

Abelardo Ruíz[(a)], David K. Anthony [(a)], Antonio Ramos [(a)]

[(a)] Dept. Ultrasonic Signals, Systems and Technology, Institute of Acoustics (CSIC),
C/ Serrano 144, 28006 Madrid. (Spain)
art@ia.cetef.csic.es, iaca344@ia.cetef.csic.es, aramos@ia.cetef.csic.es

ABSTRACT

Broadband piezoelectric transducers are used in many ultrasonic transceiver systems for Non Destructive Evaluation (NDE) and medical imaging. Precise knowledge of their internal construction parameters is required in modelling and simulation tasks to optimise behaviour in such systems. Genetic algorithms (GA) have been proposed to estimate parameters in piezoelectric backed transducers. They can help to improve the precision of the input data in the modeling and simulation tasks when this information is not available. Here, a circuital modeling approach to evaluate practical effectiveness of GA transducer estimation results, is presented and applied to achieve improvements during design and modeling tasks (with the aim of optimizing ultrasonic transceiver responses). HV spike driving and pulse-echo waveforms are simulated in time-frequency domains for selected sets of estimated parameters. Their dependences on the estimating parameters were considered, and the responses of alternative transducer designs were quantified.

Keywords: estimation parameter, genetic algorithms, transducer modeling, PSPICE simulation.

1. INTRODUCTION

Accurately determining the constructional parameters of broadband ultrasonic transducers is important in estimating the performance of pulsed transceiver systems. This type of transducer is usually involved in experimental laboratory set-ups devoted to research purposes or in the industrial context of NDE (non-destructive evaluation) of component interiors or structural materials. Often the physical internal construction details are unknown (especially for commercial probes) but parameter estimation would allow optimisation of the overall system by computer simulations based on models of the acoustic and electrical sections.

Search algorithms can be applied to estimation problems based on experimental measurements and in some cases gradient-based methods suffice. However, such methods can give misleading results if the measurements are “noisy” as can be found in ultrasonic applications of piezoelectric transducers.

A few estimation procedures for narrowband piezoelectric resonators already exist [Smits J.G., 1976. Ruíz A., Ramos A., San Emeterio J. L., 2004] but an effective and general solution for the problems related to broadband devices is still not available. A procedure to efficiently find sufficiently approximated values of design parameters in these devices would be a valuable tool to optimise their performance in pulsed ultrasonic transceivers. This can be achieved by means of computer simulations based on models of their acoustic and electrical sections behaviours.

Additionally, black-box models for transducers were utilised in [Capineri L., Masotti L., Rinieri M., and Rocchi S., 1993, Lockwood G.R., Foster F.S., 2000] for estimation where the piezoelectric and constructional parameters were not taken into account. An artificial intelligence procedure proposed for estimation of transducer parameters in a broadband piezoelectric transmitter was presented in [Ruíz A., Ramos A., San Emeterio J. L., 2004]. Here a genetic algorithm (GA) search/optimization procedure was applied to estimate four parameters related to two pulsed transmitters whose efficiency was studied in function of the number and limits of the parameters being estimated. In this preliminary work, the GA procedure was limited by its simplicity and other values of the GA parameter were not explored.

In this paper we present a more general estimation procedure using a comprehensive application of the GA to produce transducer parameter (TP) sets. This procedure is based on an extensive combination of the GA search parameters (SP) as well as a comprehensive fitness function. A metric, the euclidean distance (ED), is used to evaluate differences between estimated transducer models resulting from several GA SP-sets. Additionally, a pulse-echo circuit modeling approach is presented in order to evaluate the effectiveness of the different TP-sets obtained from the GA when implemented in a more comprehensive transceiver model. HV spike driving and pulse-echo waveforms were simulated in time-frequency domains for selected TP-sets. The differences in performance of the TP-sets resulting from the GA are analyzed and discussed.

2. MODEL BASED FUNCTION FOR TRANSDUCER PARAMETER ASSESMENT

A broadband piezoelectric transducer can be designed with a multi-element probe. This structure can be composed of an active element that realises the electro-mechanic energy conversion (the piezoceramic), one or more constructive transducer elements (such as matching layers to the propagation medium) and other elements (for example, an element added to the piezoceramic back face – usually called the backing). All, or some, of these may be included in the design model depending on the specific probe application.

For successful TP estimation the proper selection of a good model should be based on an accurate knowledge of the device internal configuration, together with an adequate metric for judging the "fitness" of each solution. These are essential. Here in the estimation procedure we have employed a piezoelectric transmitter model using a thickness-mode matrix representation in frequency domain, based on the KLM transducer equivalent circuit [Krimholtz R., Leedom F., and Matthaei G., 1970], and a seven-parameter model implementation is used here.

Four parameters are estimated: Z_{0ef} and Z_{bef} (specific acoustic impedances of the piezoceramic and the backing section), h_{33} (piezoelectric constant), v_t (piezoceramic propagation velocity). The clamped piezoceramic capacity, C_0, the nominal working frequency (f_0) and geometric area (A) have the values 1.28 x10^{-9} F, 1.093 MHz and 3.14×10^{-4} m^2 respectively. The emitting transducer performance is represented by the emission transfer function (ETF) which is considered as a function, q, of the transducer parameters:

$$ETF(\omega) = q(Z_{0ef}, Z_{bef}, h_{33}, v_t^D, C_0, f_0, A) \qquad (1)$$

Further details of the transducer model are given in [Ruíz, A., San Emeterio, J. L., Ramos, A., 2004].

3. IMPLEMENTATION OF THE GA FOR ESTIMATION

3.1. General Aspects

Genetic algorithms are stochastic-based optimisers that perform function optimisation based on the Darwinian evolution of nature [Goldberg D.E., 1989]. In general, they are applied to search spaces that are too large to be exhaustively searched or are used when the search space is multi-modal or discontinuous. Each set of TP (that define a possible solution) is coded in the form of a finite length binary "chromosome" string. Each different chromosome corresponds to a unique set of TP values. The GA is initialised with a random pool of chromosomes and consecutive generations are obtained involving to three key operations: selection, crossover and mutation. The average fitness of the generations successively increases and the process is halted after a number of generations by a suitable convergence criterion. Using an *elitist* strategy, the best-so-far solution is guaranteed to survive into the next generation.

When applying GAs, it can never normally be established whether the true global optimal solution has been found. Consecutive applications of the algorithm lead to near-optimal solutions and in most cases these out perform the existing design, and seeking the true globally optimum design is often not a necessity. A more complete description of GAs can be found in [Goldberg D.E.,1989].

As GAs rely on random numbers in its operation, the result will depend upon the start seed of the random number generator as well as of the GA SP.

3.2. The GA implementation

The success of the GA results depends on how representative the objective (or fitness) function is. It is the sum of the 'fit' between the normalised moduli and the error in the real part of the electrical input impedance, and is

$$F = \frac{1}{N}\sum_{i=i}^{N}\left[\begin{array}{c}\left(\left|ETF(\omega)\right|_{\mathrm{N}}^{\exp} - \left|ETF(\omega)\right|_{\mathrm{N_j}}^{est}\right)^2 \\ + \; k*\left((MR_{\exp} - MR_{est})/MR_{\exp}\right)^2\end{array}\right] \qquad (2)$$

$\left|ETF(\omega)\right|_{\mathrm{N}}^{\exp}$ is the normalised modulus of the experimental curve and $\left|ETF(\omega)\right|_{\mathrm{N_j}}^{est}$ is the normalised modulus of the estimated curve (from the model), where j is an index for each generation being evaluated. The normalization process is governed by $\mathrm{N_j}$ that is calculated for each generation, and applied to this same generation. This is now described.

For the first generation, the response $\left|ETF(\omega)\right|_{\mathrm{N_1}}^{\exp}$ is calculated for each TP-set in first generation. $\mathrm{N_1}$ is the maximum value over the whole generation. In subsequent generations, $\mathrm{N_j}$, is calculated similarly except the *minimum* value is evaluated.

$MR_{\exp}$ and MR_{est} are the maximum values of the real part of the impedance at the frequency of maximum resistance f_p, $\mathrm{Re}\{Z_{in}(f_p)\}$, for the experimental and GA evaluated responses. In (2), N is the number of samples in the ETF curves and k is an empirical factor introduced to regulate the influence of the second term. The GA was configured to minimise F.

The GA was implemented in MATLAB. The values of the 4 parameters (Z_{0ef}, Z_{bef}, v_t^D, h_{33},) were limited to specific ranges and coded as a binary chromosome of total length of 83 bits. The initial generation is randomly assigned. As the GA is a stochastic search algorithm its performance is dependent on the GA parameters (SP) as well as the start seed of the random number generator. So as to apply the GA in a way that does not produce results specific the SPs and the random seed number 1350 GAs were run with different combinations of SPs and start seeds.

Three values of generation size (*ncrom*) and total number of generations (*ngen*) were used each having a constant number of objective function evaluations: *ncrom*/*ngen* = {40/50, 100/20, 200/10}.

Roulette wheel selection was used in conjunction with different conditioning operations (reciprocal, linear inversion, and linear inversion with translation to zero) and scaling functions (no scaling, rank selection, Sigma scaling and logarithmic scaling). The scaling functions convert the smaller better value of the objective function to larger sections on the circumference of the roulette wheel. The scaling functions relate how much better a better solution is (e.g. on linear or logarithmic terms, for example). It is noted that when using rank selection the results are independent of the conditioning function.

Three values of crossover and mutation probability were used, $p_c = \{0.6, 0.75, 0.9\}$ and $p_m = \{0.5, 1, 2\}/ncrom$. A second crossover point was performed with a probability of 0.5 allowing both single- and double-point crossovers so that the performance of the GA is less dependent upon the order of the parameters within the chromosome.

The 1350 GA runs when processed on a desktop PC (Intel Pentium 4, 2.40 GHz) took approximately 7 hours. It is noted that there are a total of approximately 10^{25} solutions (for a 83 bit chromosome) and so an exhaustive search is not feasible (requiring more than 10^{20} years). Otherwise, methods of efficiently sampling the full combinational GA parameter set would have to be used.

4. EUCLIDEAN DISTANCE ANALYSIS

The 1350 GAs produce solutions (TP-sets) that give optimal performance (good transducer estimation). Relatively small variation in performance is seen. However, a wider variation is seen in the TP-sets. The Euclidean distance (ED) was used as a metric to measure the distinctness between TP-sets, effectively indicating the difference in the resulting transducer design. This is not the same as differences in the value of their objective functions (i.e., the performance). Using the ED, transducer designs that are distinct can be sought and the differences between them studied.

For *M*-parameter model the *ED* between two solutions, *r* and *s*, is calculated from the normalised differences in each parameter, *m*, denoted as $a_m(r)$ and $a_m(s)$, and is

$$ED(r,s) = \sqrt{\frac{1}{M}\sum_{\forall m}\left(\frac{a_m(r) - a_m(s)}{a_{m,\max} - a_{m,\min}}\right)^2} \qquad (3)$$

$a_{m,min}$ and $a_{m,max}$ are the minimum and maximum values of parameter *m* used in the GA. The *ED* measure has the range {0,1}.

Z_{0ef}	Z_{bef}	h_{33}	v_t^D	F	ED	SP-set
3.41E6	3.69E7	1.92E9	4036	5.6E-1	.232	*BS*
2.53E6	2.89E7	1.72E9	3615	7..9E-1		*AS*

Table 1.TP-sets, F and ED values of *BS* and *AS*.

The EDs between the TP-sets resulting from the 1350 SP-sets were calculated. This enables many different cases to be studied but here only one is presented. The best overall TP-set (denoted *BS*) (with the lowest value of *F*) was compared with another TP-set (denoted *AS*) having a very similar value of *F* but with distinct TPs. In this case the ED was greater than 0.2. Table 1 shows TPs and values of *F* for the two transducer designs.

5. EVALUATING THE EFFECTIVENESS OF THE GA ESTIMATION USING A CIRCUITAL PULSE-ECHO MODEL

A circuital pulse-echo (P-E) model was employed in order to evaluate by simulation the effectiveness of the GA estimation procedure [A. Ruíz, M. Hernández, A. Jiménez, A. Sotomayor, O. Sánchez, A. Ramos , P. T. Sanz, J. L. San Emeterio, 2002]. This emitting-receiving arrangement was implemented in PSPICE [Orcad^{TM}]. Figure 1 presents the circuit diagram of the wideband P-E arrangement used.

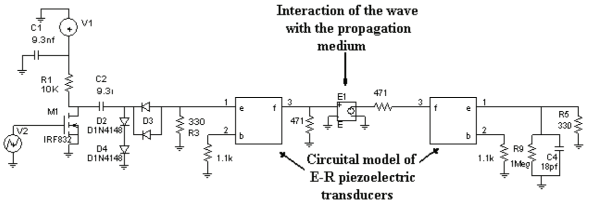

Interaction of the wave with the propagation medium
C1
9.3nf
V1
R1
10K
C2
9.3
M1
V2
IRF832
D2
D1N4148
D3
D4
D1N4148
330
R3
1.1k
1
2
e
f
b
3
471
E1
471
3
f
e
b
1
2
1.1k
R9
1Meg
C4
18pf
R5
330
Circuital model of E-R piezoelectric transducers

Figure 1: Electrical diagram of the ultrasonic Pulse-Echo arrangement.

The emitter and receiver (E-R) transducers are represented by identical three-port blocks based on a Redwood equivalent circuit [Redwood M., 1961]. These included a quadratic frequency dependence of the mecanical losses in the piezoceramic material [Ramos A., Ruíz A., San Emeterio J. L., Sanz P.T., 2006]. The emitter and receiver blocks represent the two functions (transmit and receive) of one physical transducer separately and have identical geometric and piezoelectric parameters. Pin “e” represents the electrical port, and “b” and “f” denote the back and front mechanical ports of the probes.

The electronic driving stage was modelled by a simplified configuration of a pulsed generator circuit as those employed in NDE. The receiving stage effects were considered by means of a simplified receiver input impedance. Some aspects of the input impedance of the oscilloscope and a damping resistor of the driving structure were included. The resistor is a first approximation of the excitation circuit effects in the echografic signal.

Block E1 represents a dependent voltage source which electrically decouples both piezoelectric stages. From the acoustical point of view E1 models (in a very simplified way) the interaction of the acoustic wave with the propagation medium. The basic gain, β, of E1 can be expressed by the following equation:

$$\beta = 2\,e^{-\alpha x}\left(\frac{Z2 - Z1}{Z2 + Z1}\right) \qquad (4)$$

where α is the attenuation cofficient of the propagating wave and *x* is the distance travelled by the wave in the propagation medium. The fraction in equation (4) is the reflection coefficient at the interface reflector material - propagation medium, where Z1 and Z2 are the acoustic impedance of the propagation medium and reflector material respectively.

With minor changes, the circuit diagram in Figure 1 can be also employed to model Through - Transmission (T-T) responses. In this case, the ratio between the geometric areas of the receiver and emitter transducers must be introduced in (4). Additionally, an inductive matching component can also be included in the emitting -receiving electronic to model other T-T cases.

6. RESULTS

6.1. From the GA estimation procedure and the Euclidean distance employment

Figure 2a) shows a comparison between the experimental (in blue) and two estimated curves of the normalized modulus of the ETF for an estimation process of four parameters. The black curve represents the *BS* taking into account the equation (2). The green curve was obtained for the TP-set *AS*. All curves have been normalized because the fitness function is based on the shape of the ETF modulus and Re{$Z_{in}(f_p)$}, where this last curve takes account of the amplitude aspects.

From the transducer estimated parameters, the electrical impedance has been computed and the real part of this transducer characteristic function is presented in Figure 3b). A reasonably good agreement can be appreciated in all the curves shown in Figure 3. Nevertheless, the *BS* has better agreement with Re{Z_{in}} than *AS*, although the difference is small.

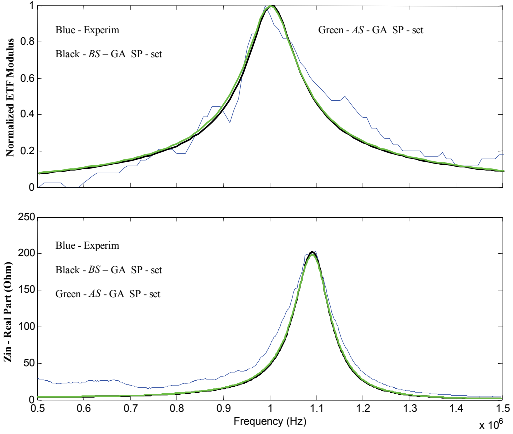


Figure 2: Comparison between the experimental and two GA estimated curves versus the frequency: a) Normalized Modulus of the ETF b) Real part of the electrical input impedance.

Here, only Re{Z_{in}} is shown because this is more directly related with the second term of the fitness function used in this work. However, it is noted that the imaginary part of the computed Z_{in} also has reasonably good agreement with the experimental curve but is not presented here.

6.2. Employment of the circuital modeling approach

Based on the configuration presented in Figure 1 simulated P-E results were obtained for a reflection of the emitted ultrasonic pulse in an interface water/plastic located at a near-field distance of 50 mm from the transducer front face. The reflector material considered was PMMA (with a specific acoustic impedance, Z_{PMMA}, of 3.2 MRayls).

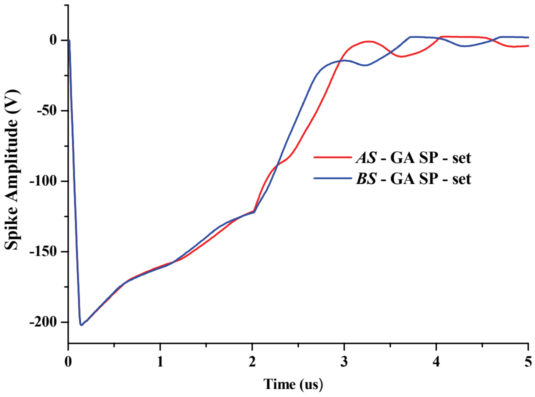


Figure 3: Driving responses for the P-E configuration with a piezoceramic transducer modelled from TP sets obtained from two different GA SP-sets: Blue: *BS* - SP-set. Red: *AS* - SP-set.

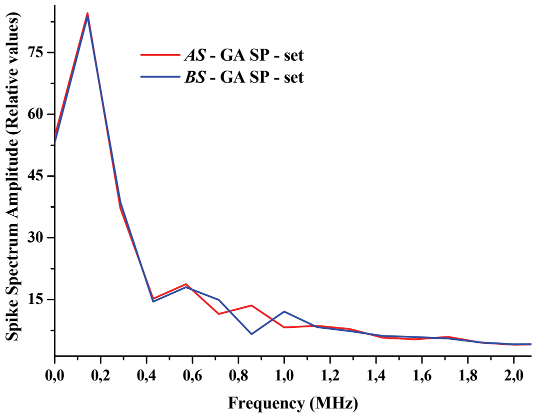


Figure 4. Driving frequency spectra for the pulse-echo configuration involving a piezoceramic transducer modelled with TP-sets obtained from two different GA SP-sets: Blue: *BS* - GA SP-set. Red: *AS*-GA SP- set.

Figure 3, presents the simulated driving temporal responses for the P-E configuration for a piezoceramic transducer modelled with TP-sets obtained from *BS* and *AS* SP-sets. A slightly shorter rise time for the *AS* TP-set can be seen. Additionally, there are certain distortions in the spike waveforms that are induced from the motional behavior of the piezoelectric loads. These are more clearly appreciated for *AS*. The differences in both simulated signals can be influenced by the different values of the estimated impedances associated to every transducer (see Table 1) [Ramos A., San Emeterio J. L., Sanz P.T, 2000].

In Figure 4, a comparison of the driving frequency spectra for the P-E configuration with a piezoceramic transducer modelled with *BS* and *AS* TP-sets is made. Here, the colors of the curves are the same as those in Figure 3. The behaviour of the spike waveforms is very similar, especially at low frequencies. Nevertheless, in the frequency range of interest to ultrasonic responses (0.5-1.5 MHz), some notable differences between both curves can be appreciated.

Two simulated pulse-echo waveforms obtained with *BS* and *AS* TP-sets are shown in Figure 5. The blue line shows the results obtained with the *BS* TP-set. The red line shows the results with the *AS* TP-set. There is a notable difference between the amplitudes of both waveforms, as well as a small difference in the resonance frequency. These disagreements can be caused by the difference of some of the estimated transducer parameters, among which is the longitudinal wave propagation velocity in the piezoceramic which directly influences the transducer frequency. On the other hand, the temporal pulse length is shorter for the *BS* option. This can be explained by the higher value of Zb than the *AS* TP-set.

A comparison of the pulse-echo frequency responses associated with the curves presented in Figure 5, is shown in Figure 6. The frequency shifting can be more clearly apreciated in the resonance peaks of the second harmonic.

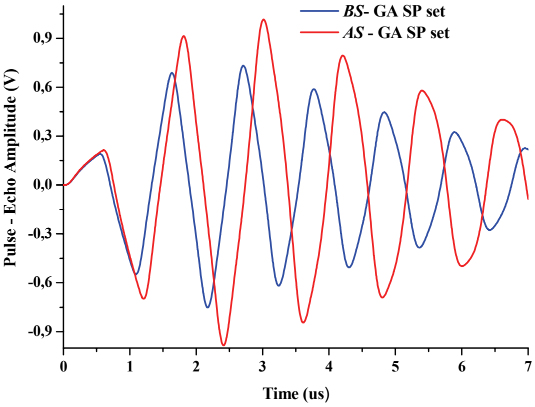


Figure 5. Simulated pulse-echo temporal responses. Blue color: *BS* GA SP- set. Red color: *AS* GA SP - set.

The temporal pulse length of the pulse emitted by the transducer can be influenced by the characteristics of the backing element. The backing element is a high density and highly attenuative material that is used to absorb the energy radiating from the back face of the piezoceramic, and controls its vibration. This pulse length is closely related with the transducer bandwidth. The shorter the pulse duration (or length), the larger the bandwidth of the transducer and this can be achieved by the backing effect.

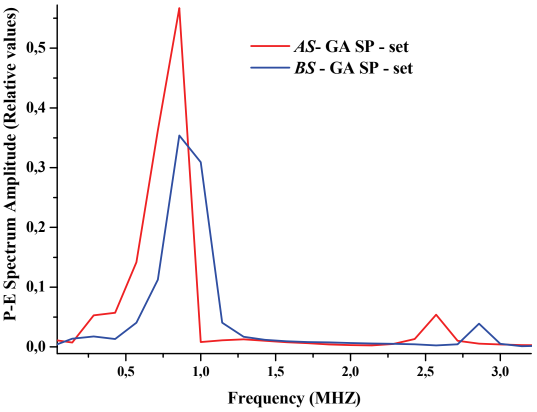


Figure 6: Simulated pulse-echo frequency responses. Blue color: *BS* GA SP- set. Red color: *AS* GA SP-set.

7. SUMMARY AND CONCLUSIONS

Improvements in GA procedure for estimation of constructional parameters in wideband piezoelectric transducers have been presented using a model based on the matrix representation of the piezoelectric transmitter topology. The procedure has been applied to the estimation of four unknown internal parameters, considering up to 1350 possible designs resulting from different GA parameters.

A metric, based on the euclidean distance, was introduced and applied in order to analyze different ultrasonic responses between groups of estimated parameters associated to GA SP sets selected by means this index.

A pulse-echo (P-E) circuital modeling approach to evaluate practical effectiveness of the different sets of solutions, obtained from the GA procedure and the metric employment, have been presented and applied.

The analysis of several P-E simulated responses in time-frequency domains, allowed clear differences between models with different TP-sets to be established from the point of view of the transducer design, although having similar estimated performance.

The neccesity and usefulness of complementing the analysis of the different GA estimation solutions with modelling and simulation tasks was shown. This can help to achieve a better understanding of some physical phenomena involved in the design and functioning of the wideband ultrasonic transducers.

ACKNOWLEDGMENTS

This work was supported by the R&D National Plan of the Spanish Ministry of Education & Science (Project PN DPI2005-00124).

REFERENCES

Smits J.G., 1976. Iterative method for accurate determination of the real and imaginary parts of the materials coefficients of piezoelectric ceramics, *IEEE Trans. on Son. and Ultrason.*, SU-23 (6), 393-402

Ruíz, A., San Emeterio, J. L., Ramos, A., 2004. Evaluation of piezoelectric resonator parameters using an artificial intelligence technique. *Integrated Ferroelectrics***,** 63, 137-141.

Capineri L., Masotti L., Rinieri M., and Rocchi S., 1993, Ultrasonic transducer as a black-box: Equivalent circuit synthesis and matching network design. *IEEE Trans. Ultrason., Ferroelect., Freq. Cont.*, 40 (6), 694-703.

Lockwood G.R., Foster F.S., 2000. Modeling and optimization of High-Frequency ultrasound transducers. *IEEE Trans. Ultrason., Ferroelect., Freq. Cont.*, 41 (2), 826-835

Ruíz A., Ramos A., San Emeterio J. L., 2004. Estimation of some transducer paramaters in a broadband piezoelectric transmitter by using an artificial intelligence technique, *Ultrasonics***,** 42, 459-463.

Goldberg D.E., 1989.Genetic Algorithms in Search, Optimization and Machine Learning. Cambridge, MA: Addison-Wesley

Krimholtz R., Leedom F., and Matthaei G., 1970. New Equivalent Circuits for Elementary Piezoelectric Transducers, *Electronics Letters*, 6 (13), 398-399

Orcad^{TM}, Beaverton, OR USA. Available from: (www.orcad.com)

Redwood, M. 1961. Transient Performance of a Piezoelectric Transducer. *J. Acoustic. Soc. Amer*., 33 (4), 527-536.

Ramos A., Ruíz A., San Emeterio J. L., Sanz P.T., 2006. PSpice Circuital Modelling of Ultrasonic Imaging Transceivers including Frequency-dependent Acoustic Losses and Signal Distortions in Electronic Stages, *Ultrasonics*, 44, e995–e1000.

Ruíz A., Hernández M., Jiménez A., Sotomayor A., Sánchez O., Ramos A., Sanz P. T. , J. L. San Emeterio, 2002. Approaches to simulate temporal responses in piezoelectric stages involved in pulsed ultrasonic sensor/actuator systems, *Ferroelectrics*, 273, 243-248.

Ramos A., San Emeterio J. L., Sanz P.T, 2000. Dependence of pulser driving responses on electrical and motional characteristics of NDE ultrasonic probes, *Ultrasonics*, 38, 553-558.

A NEURAL MAXIMUM SELECTOR: EXPLICIT PARAMETER SET-UP FOR TIME PERFORMANCE

Ruxandra L. Costea[(a)], **Corneliu A. Marinov**[(b)]

[(a)]Polytechnic University of Bucharest, Romania
[(b)]Polytechnic University of Bucharest, Romania

(a)rux_co@itee.elth.pub.ro, (b)cmarinov @rdslink.ro

ABSTRACT

A continuous time neural network of Hopfield type is considered. It is a W(inner) T(akes) A(ll) selector. Its inputs are capacitively coupled to model the parasitics or faults of overcrowded chip layers. A certain parameter setting allows the correct selection of the maximum element from an input list. As processing time is a performance criterium, we infer upper bounds of it, explicitly depending on circuit and list parameters. Our method consists of converting the system of nonlinear differential equations describing the circuit to a system of decoupled linear inequalities.

Keywords: neural networks, winner-takes-all, Hopfield networks, parasitics, time evaluations.

1. INTRODUCTION

We consider a neural network of N cells with a complete interconnection of negative feedback type. We design it as a WTA machine, i.e. a mean of separating the largest signal from a list of constant and distinct signals. This is an up to date topic in time problems for neural networks of analog type (Wu 2001, Cao 2001, Zhang, Heng and Fu, 1999; Liang and Si 2001, Cho 2005). It follows the pioneering papers regarding the computational Hopfield networks (Hopfield 1984, Hopfield and Tank 1985, Majani, Erlanson and Abu-Mostafa 1989; Atkins 1992, Dranger and Priemer 1997).

Referring to Fig. 1 each cell is an ideal amplifier with $v_i = mg(\lambda u_i)$ where g is a “sigmoid” i.e. $g:\Re \to (-1,1)$, $g'(x) \geq a > 0$, $\lim_{x\to\pm\infty} g(x) = \pm 1$ and $\lim_{x\to\pm\infty} xg'(x) = 0$. $\lambda > 0$ is “the gain”. Apart from interconnections $p > 0$ we consider here the mutual capacitance δ between all pairs of inputs. It models the unavoidable parasitic effects on the crowded chips. We are interested in the influence of these capacitances on the network performances: is the WTA selection still working? How much is the network speed affected?

The processing list is fed by current sources d_i ordered as:

$$d_{\sigma(1)} > d_{\sigma(2)} > \cdots > d_{\sigma(N)} \qquad (1)$$

where $\sigma(i)$ is a permutation of indices 1 to N. The network should signal that $d_{\sigma(1)}$ is the “winner”. This is done by choosing proper values of circuit parameters M, λ, m, a, p, l including the $[0, d_{\max}]$ admission interval and t_p, the clocking time. Also we have to take into account the minimum density z of arriving sequence of lists, which is $z = \dfrac{\Delta(N-1)}{d_{\max}}$ where $\Delta = \min|d_i - d_j|$.

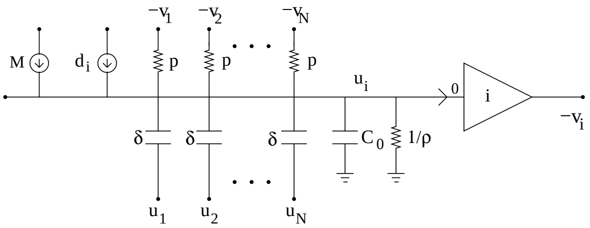


Figure. 1: The i-th cell with all its interconnections

The network in Fig. 1 is described by

$$C\frac{\mathrm{d}u}{\mathrm{d}t} = -lu - Tv + b \qquad (2)$$

where $u = (u_1, \cdots u_N)^{tr}$, $v = (v_1, \cdots, v_N)^{tr}$, $b = (b_1, \cdots, b_N)^{tr}$ with $b_i = d_i + M$. C is the capacitance matrix with $C_{ii} = C_0 + (N-1)\delta$ and

$C_{ij} = -\delta$. T is the resistive interconnection matrix, $T_{ii} = 0$, $T_{ij} = p > 0$. Also $l = \frac{1}{\rho} + (N-1)p$.

As we show elsewhere, for each initial condition (2) has a unique solution $u(t)$ defined on $[0,\infty)$. Also, for almost all vector sources $b \in \Re^N$ (2) has a finite number of equilibria $\bar{u}$. They are solutions of the stationary equation

$$0 = -lu - Tv + b \tag{3}$$

We can also show that the old Liapunov function introduced by Hopfield (Hopfield 1984), namely

$$E(u) = \frac{1}{2}\langle Tv, v\rangle - \langle b, v\rangle + \frac{l}{m\lambda}\sum_{i=1}^{N}\int_0^{v_i} g^{-1}\left(\frac{x}{m}\right)dx$$

works for our special case of C being non-diagonal. This implies that for any solution $u(t)$ of (2) there exists an equilibrium $\bar{u}$, solution of (3), such that $u(t) \to \bar{u}$ when $t \to \infty$.

The proofs are omitted below. They can be found by the methods in (Calvert and Marinov 2000, Marinov and Calvert 2003, Marinov and Hopfield 2005, Chen, Lu and Amari 2002; Costea 2007, Costea and Marinov 2006, Costea and Marinov 2007).

2. THE WTA SELECTOR

Here we give conditions on circuit and list parameters such that once the list (1) arrives, the circuit evolves toward an equilibrium $\bar{u}$ with

$$\bar{u}_{\sigma(1)} > \beta > -\beta > \bar{u}_{\sigma(i)} \tag{4}$$

for all $i \in \overline{2, N}$. Here β is a threshold assuring a convenient resolution of output.

The first result we give is "the ordering" property of the dynamic solution. This is, starting from zero and imposing $\lambda > \frac{l}{pam}$ the order in (1) (where we took $\sigma(i) = i$ for writing simplicity) transfers to $u(t)$:

$$u_1(t) > u_2(t) > \cdots u_N(t) \tag{5}$$

for all $t > 0$. Then, by using "the difference equation"

$$0 = -l(\bar{u}_i - \bar{u}_{i+1}) + p(\bar{v}_i - \bar{v}_{i+1}) + d_i - d_{i+1} \tag{6}$$

and the above convergence, we derive

$$\bar{u}_1 > \bar{u}_2 > \cdots \bar{u}_N \tag{7}$$

Next, we can show the WTA property

$$\bar{u}_1 > \beta > -\beta > \bar{u}_2 > \bar{u}_3 > \cdots > \bar{u}_N \tag{8}$$

provided that the following conditions are met:

$$\Delta \geq 2l\beta \tag{9}$$

$$M \geq l\beta - p(N-1)\xi - \underline{d}_1 \tag{10}$$

$$M \leq -l\beta + p\xi - p(N-2)m - \bar{d}_2 \tag{11}$$

Here $\xi = mg(\lambda\beta)$, $\underline{d}_1 = zd_{\max}$ is the lowest d_1 and $\bar{d}_2 = d_{\max} - (N-2)\Delta$ is the highest d_2. Thus (9)-(11) give conditions for (8) regardless the lists with density bigger than z.

3. TIME BOUNDS

We try now to obtain a clocking time for our machine. The moment t_p when we should stop the transient of $u(t)$ towards $\bar{u}$ is when the WTA property (8) of $\bar{u}$ is fulfilled:

$$u_1(t_p) > \beta > -\beta > u_2(t_p) > \cdots > u_N(t_p) \tag{12}$$

As t_p is unknown, we try to obtain an upper bound of it T_p at which (12) is still valid.

We distinguish two cases – Figs 2 and 3.

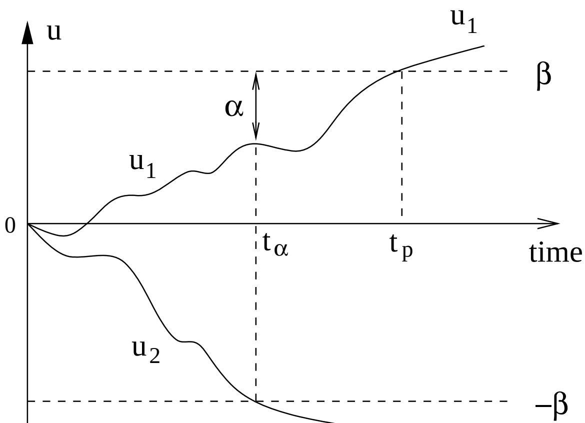


Figure. 2 The processing phase - case 1. The $u_{\sigma(1)}$ winner surpasses the threshold β after the moment when the losers $u_{\sigma(2)}$ fall under $-\beta$

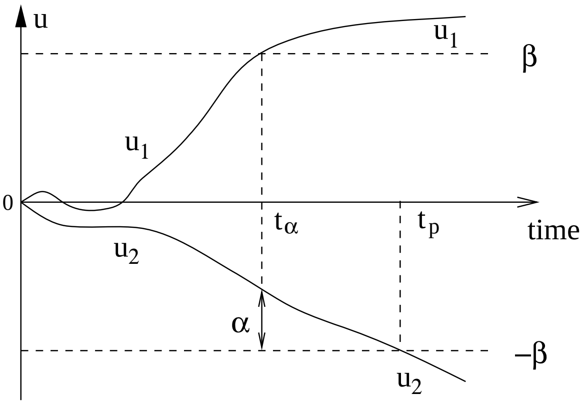


Figure. 3 The processing phase - case2. The $u_{\sigma(1)}$ winner goes above β before the moment when $-\beta = u_{\sigma(2)}$.

The first one, supposes $u_2(t)$ passes the WTA threshold $-\beta$ before $u_1(t)$ crosses its $+\beta$ level. Rigorously speaking, we suppose $\alpha \in [0,2\beta]$ and take the moment t_α when $u_1(t_\alpha) = \beta - \alpha$, $u_2(t_\alpha) = -\beta$ and for all $t > t_\alpha$, $u_2(t) < -\beta$. The second case supposes $u_1(t)$ reaches $+\beta$ in advance of $u_2(t)$ touching $-\beta$. In this case we call t_α the moment when $u_1(t_\alpha) = \beta$, $u_2(t_\alpha) = -\beta + \alpha$ and for all $t > t_\alpha$ $u_1(t) > \beta$. In both cases above we call t_p -processing time-, the first moment after t_α when $u_1(t_p) \geq \beta$ and $u_2(t_p) \leq -\beta$. With these, the problem of finding the clocking time T_p reduces to search for upper bounds $\bar{t}_\alpha$ and $\overline{t_p - t_\alpha}$ of t_α and $t_p - t_\alpha$ respectively. We have $T_p(\alpha) = \bar{t}_\alpha + \overline{t_p - t_\alpha}$.
The first bound $\bar{t}_\alpha$ comes from “the difference equation”

$$C_n \frac{\mathrm{d}}{\mathrm{d}t}(u_1 - u_2) = -l(u_1 - u_2) + p(v_1 - v_2) + d_1 - d_2$$

where $C_n = C_0 + N\delta$. With $(u_1 - u_2)(0) = 0$ and $(u_1 - u_2)(t_\alpha) = 2\beta - \alpha$ we get

$$t_\alpha \leq \bar{t}_\alpha = \frac{C_n}{l} \ln \frac{\Delta}{\Delta - l(2\beta - \alpha)} \tag{13}$$

This is valid equally for the two cases and all $\alpha \in [0,2\beta]$.

The evaluation of $t_p - t_\alpha$ in case 1 comes from the first equation in (2) written as:

$$C_0 \frac{\mathrm{d}u_1}{\mathrm{d}t} = -lu_1 - \delta \sum_{j=1}^{N} \frac{\mathrm{d}}{\mathrm{d}t}(u_1 - u_j) - p\sum_{j=2}^{N} v_j + b_1$$

It yields

$$t_p - t_\alpha \leq \overline{t_p - t_\alpha} = \frac{C_0}{l} \ln \frac{W - l\beta + l\alpha}{W - l\beta} \tag{14}$$

where

$$\begin{aligned} W &= p\xi(N-1) + \underline{d}_1 + M - \\ &- \frac{\delta}{C_n}[2pm(N-1) + \sum_{j=1}^{N} \bar{d}_{1j}] \end{aligned} \tag{15}$$

Here $\underline{d}_1 = zd_m$ and $\bar{d}_{ij} = d_m - (N-j)\Delta$.
For the case 2 we use the second equation in (2)

$$C_0 \frac{\mathrm{d}u_2}{\mathrm{d}t} = -lu_2 - \delta \sum_{j=1}^{N} \frac{\mathrm{d}}{\mathrm{d}t}(u_2 - u_j) - p\sum_{\substack{j=1 \\ j\neq 2}}^{N} v_j + b_2$$

and get again (14) where

$$\begin{aligned} W &= p\xi - pm(N-2) - \bar{d}_2 - M - \\ &- \frac{\delta}{C_n}\left[2pm(N-1) + \bar{d}_{12}\right] \end{aligned} \tag{16}$$

Here $\bar{d}_2 = d_m - \Delta$ and $\bar{d}_{12} = d_m - (N-2)\Delta$.
Now, (13) and (14) give the bound $T_p(\alpha)$ of processing time for every $\alpha \in [0,2\beta]$. By imposing $\frac{\mathrm{d}T_p}{\mathrm{d}\alpha} < 0$ we find $\max T_p(\alpha) = T_p(0)$ which gives a final bound:

$$t_p \leq T_p(0) = \frac{C_n}{l} \ln \frac{\Delta}{\Delta - 2l\beta} \tag{17}$$

The above imposition results in

$$W - l\beta > 0 \tag{18}$$

for both two cases, and also $\Delta - 2l\beta > 0$ as in (9). These inequalities are made true by a proper choosing of circuit parameters M, ξ, β, m, p, d_m when the minimum list density z and the maximum parasitic capacitance δ are given. Our evaluations works for $\delta \in [0, C_0/N - 2]$.

Also, in this context we can answer the natural question: "is the processing time longer when the parasitic capacitance increases?" For these, by knowing from above that the maximum of t_p happens when u_1 and u_2 simultaneously reach $+\beta$ respectively $-\beta$, we give bilateral bounds of t_p:

$$\frac{2p\xi + d_{12}}{2p\xi + d_{12} - 2l\beta} \le e^{\frac{l}{C_n} t_p} \le \frac{d_{12}}{d_{12} - 2l\beta} \tag{19}$$

If t_p^{δ} is the processing time with parasitic capacitance δ then the from (19) we can easily infer $t_p^{\delta} \ge t_p^{\delta=0}$ for $\delta > \frac{2pm}{N(\Delta - 2l\beta)} C_0$.

4. CONCLUSION

All of above give analytical relations between parameters to fulfill the WTA property. They are (9)-(11) and (18). The clocking time is given by (17), which provides a mean to influence the processing speed when the crosstalk is considered and very tight lists are fed. The assumptions under which our results are reasonable for practical purposes.

REFERENCES

Wu, J., 2001 *Introduction to neural dynamics and signal transmission Delay*, Berlin, NY, Walter de Gruyter,.

Gupta M.M., Jin L. and Homma N., 2003, *Static and Dynamic Neural Networks from Fundamentals to Advanced Theory*, Hoboken, IEEE Press, Wiley-Interscience.

Hopfield J.J., 1984, Neurons with graded response have collective computational properties like those of two state neurons, *Proc. Nat. Academy Sci.*, Vol. 81, pp. 3088-3092.

Hopfield J.J. and Tank D. W., 1985, Neural computation of decisions optimization problems, *Biol. Cybern.*, vol. 52, pp. 141-152.

Atkins M., 1992, Sorting by Hopfield nets, *Proc. Int. Joint Conf. Neural Networks*, vol. 2, pp. 65-68, Washington DC.

Dranger T.S. and Priemer R., 1997 Collective process circuit that sorts, *IEE Proceedings-Circuits, Devices, Syst*, vol. 144, pp. 145-148.

Cao J., 2001, Global exponential stability of Hopfield neural networks, *Int. J. Syst. Sci.*, Vol. 32, no. 2, pp. 233-236.

Majani E., Erlanson R. and Abu-Mostafa Y., 1989, On the K-winner-take-all network, *Advances in Neural Information Processing Systems, D.S. Touretzky, Ed. San Mateo, CA: Morgan-Kafmann*, vol.1, pp. 634-642.

Zhang Y., Heng P.A. and Fu W. C., 1999, Estimate of exponential convergence rate and exponential stability for neural network, *IEEE Trans. Neural Networks*, vol. 10, no. 6, pp. 1487-1493.

Liang X. and Si J., 2001, Global exponential stability of neural networks with globally Lipschitz continuouns activations and its application to linear variational inequality problem, *IEEE Trans. on Neural Networks*, Vol. 12(2), pp. 349-359.

Chen T., Lu W. and Amari S., 2002, Global Convergence Rate of Recurrently Connected Neural Networks, *Neural Computation*, vol. 14, pp. 2947-2957.

Calvert B. and Marinov C. A., 2000, Another K-winners-take-all analog neural network, *IEEE Trans. Neural Networks*, Vol.11, pp. 829-838.

Marinov C. A. and Calvert B. D., 2003, Performance analysis for a K-Winners-Take-All analog neural network: Basic theory, *IEEE Trans. Neural Networks*, Vol. 14, pp. 766-780.

Cho K., 2005, Delay calculation capturing crosstalk effects due to coupling capacitors, *Electronic Letters*, Vol. 41, no. 8, pp. 458-460.

Marinov C. A. and Hopfield J. J., 2005, Stable computational dynamics for a class of circuits with $O(N)$ interconnections capable of KWTA and rank extractions, *IEEE Trans. Circuits and Systems, Part 1*, Vol. 52, pp. 949-959.

Costea R. L., 2007, *Artificial neural systems - Switching times for WTA circuits of Hopfield type*, Doctoral Disertation, Polytechnic University of Bucharest.

Costea R. L and Marinov C.A., 2006, Capacitive cross-coupling faults and WTA correct behaviour, *Proc of 10th IEEE Workshop on Signal Propagation on Interconnects*, pp. 189-192, Berlin, Germany.

Costea R. L and Marinov C.A., 2007, Clocking and WTA design of a continuous time Hopfield net with parasitic capacitances, *European Conference on Circuit Theory and Design*, Seville, Spain.

AUTHORS BIOGRAPHY

Ruxandra L. Costea received the B.Sc. in Electronic Engineering, the M.Sc and the Ph.D. in Electrical Engineering all from Polytechnic University of Bucharest, Romania in 2000, 2001 and 2007 respectively. She serves now as a lecturer with the Department of Electrical Engineering. Dr. Costea works in Neural Network dynamics, Nonlinear Analog Circuit and Systems, Computational Intelligence. She is an IEEE member. She presented her results to many prestigious international meetings as ECCTD – Seville 2007, ICCNS – Boston 2008, SPI – Berlin 2006, MIXDES – Gdynia 2006, ECCOMAS – Jyvaskyla 2004, ICSRIC – Baden-Baden 2001.

Corneliu A. Marinov received the Ph.D. degree in Electrical Engineering from Polytechnic University of Bucharest, Romania, in 1977, the M.Sc. degree in mathematics from the University of Bucharest in 1981 and the Docent degree in applied mathematics from University of Jyvaskyla, Jyvaskyla, Finland, in 1990. Since 1970, he has been with the Electrical Engineering Department of Polytechnic University. Meanwhile, he has held temporary positions or lectured as Visiting Professor at the University of Jyvaskyla, the University of Manitoba, Winnipeg, MB, Canada, the University of Texas at Arlington, Arlington, Middlebury College, Middlebury, VT, the Polytechnic University of Catalunia, Barcelona, Spain, and the State University of New York, Stony Brook. He has done research on theory of nonlinear circuits and systems, RC circuit transients, waveform relaxation, delay time estimation for digital interconnections, mixed type circuits with distributed and lumped parameters, monotone operators as models for circuits, and neural network dynamics. He has coauthored a book, *Mathematical Models in Electrical Circuits, Theory and Applications* (Boston, MA: Kluwer, 1991) and he has published 80 papers. Dr. Marinov is an IEEE Senior Member and served as an Associate Editor for *Circuits, Systems, and Signal Processing.* He was a Fulbright scholar and a Rockefeller Foundation grantee. He is listed in Who’s Who in the World.

OPTIMIZING HINTERLAND TERMINAL OPERATION USING SIMULATION AND NEURAL NETWORKS

Gronalt Manfred [(a)], Benna Thouraya [(b)], Posset Martin [(c)]

[(a)] [(b)] [(c)] University of Applied Life Sciences, Vienna. Department of Economics and Social Sciences, Production and Logistics. Feistmantelstrasse 4, A-1180, Vienna, Austria.

[(a)] manfred.gronalt@boku.ac.at., [(b)] thouraya.benna@boku.ac.at, [(c)] martin.posset@boku.ac.at

ABSTRACT

In this paper we investigate a major problem in hinterland terminal optimization. Terminal operation consists of a series of interdependent activities and decision problems. The overall performance in the terminal operation is influenced by operation's efficiency and differs for different terminal design, workload and policy. We use a methodology that combines simulation and neural networks and that can be used to define, with respect to terminal design and workload, the best operating policy.

Keywords: Simulation, Neural Networks, Hinterland Terminals, Optimization.

1. INTRODUCTION

Hinterland terminals enable the transhipment of load units between various modes of transport (ship, truck and train) and play a significant role in intermodal freight transport. Since intermodalism in general has become an important issue (Bontekoning, Macharis, and Trip 2004), hinterland terminals are looking to increase their effectiveness and efficiency.

Terminal operation consists of a series of interdependent activities which describe the container flow through the terminal. An overview of container terminal operation, involving ship, train and truck transport, is given by Vis and de Koster (2003).

Terminal activities take place in three major areas: the interchange area where transport modes enter the terminal, the transhipment area where the loading and unloading is done and the yard area where load units are stored. The overall performance in terminal operation is influenced by the operation's efficiency of all these areas. While optimizing terminal operation, one therefore, needs to take into consideration all existing interdependencies. For example, the scheduling problem for loading and unloading activities of a train is highly related to the storage allocation problem in the yard and to the track allocation problem of the rail interchange.

Further, terminal optimisation has to take into account decisions of different time horizon (Meersmans and Dekker 2001). While strategic decisions focus on terminal design, tactical and operational decisions deal with operating policies, as the assignment and scheduling of terminal resources. Defining layout and equipment of a terminal has therefore a direct impact on the efficiency of the chosen policies. Moreover, the decision on which policy to choose is highly influenced by the predefined terminal design.

Finally, terminal performance is also affected by the actual workload. Features as arrival pattern, average storage time, load unit characteristics or modal split determine, in combination with terminal design, the utilization degree of the resources. Different combinations of terminal design and workload (denoted as terminal configuration) can in fact have different impacts on terminal performance.

Due to the complexity of terminal operation and to the various existing interdependencies a systematic approach is needed. In this paper we propose a method that can be used to determine, with respect to the predefined configuration, which policy would result in a favourable terminal performance.

Most available research focuses on one specific terminal area or decision problem, allowing thus only for partial optimization (Steenken, Voß, and Stahlbock 2004). Some research, mostly simulation based, is done on the overall performance of container terminals. In fact computer-based simulation is particularly apt to describe the inner workings of a terminal (Rizolli, Fornara, and Gambardella 2002) and can therefore be used to assess the impact of a specific configuration and policy on terminal performance. Because simulation is mainly used to analyze the outcome of predefined scenarios, fundamental insights into factors affecting terminal performance are still lacking.

The method that we propose combines simulation with optimization and outlines the functioning of container terminal systems. Due to the great number of parameters and constraints describing the terminal operation, we do not want to explicitly define and explain the existing interdependencies. In fact, we choose to approximate the function linking terminal configuration and policy to terminal performance by implementing a neural network. The neural network uses simulation results from scenario analyses to estimate a non-linear function representing the terminal system.

2. PROBLEM STATEMENT

Terminal managers face a complex decision making environment where a large set of strategic, tactical and operational decisions have to be solved.

One of the most crucial decisions is related to allocation of storage capacities to the incoming containers. In fact, storage space is in most European hinterland container terminals a scarce good, and has therefore to be used efficiently. From the point of view of mere terminal processes, the storage area is used to bridge the time gap between arrival and departure time of a container. When a container is delivered for example by a train and the picking truck has not yet arrived, the container has to be lifted into the yard. This additional lift is necessary as the train has a limited time for the unloading process. The yard manager has therefore to make sure that for all arriving containers, an adequate storage space is reserved. Knowing that terminal operators often take considerable profits from the storage fees charged, it becomes clear that a balancing decision has to be solved.

Further, the goal of the yard manager is not only to provide for storage space, but also to do this efficiently. This means that the storage movement has to be done as fast as possible, to reduce the transport mode waiting time. Therefore, the storage allocation decision has to take into account the best equipment allocation and the best storage allocation. This means that the transport and lifting time of the container has to be optimized by allocating the nearest handling equipment and choosing the nearest storage spot which simultaneously reduces future unproductive moves. Whereas unproductive moves are defined as reshuffles, which are required to access another container that is stored beneath it. This implies that reshuffles occur only when removing containers from the stack.

Reducing the number of unproductive moves and minimizing the travel distance of the handling equipment consequently improves the container movement time, which reduces the residence time of the transport mode. A higher utilization rate of the yard however results in longer handling times, as the number of available storage spots is reduced. The main goal of the yard manager is therefore, to choose a storage strategy which matches the current terminal circumstances best.

During on-field visits and interviews with experienced yard manager, we observed several storage policies.

1. Avoid container stacking as long as the yard utilization allows for. If stacking is unavoidable choose from policy 2 to 4.
2. Choose nearest available storage slot without considering any stacking constraint
3. Choose nearest available storage slot while avoiding to stack on a container with an earlier expected pick-up time.
4. Prefer container stacking, while stacking only import containers with same arrival time and train number. Containers delivered by truck stock according to policy 3.
5. Prefer stacking and group all containers according to their destination and source.
6. Choose from policy 3 to 5 and segregate storage area according to container dwell time characteristics (storage or transshipment container).
7. Choose from policy 3 to 5 and separate import and export containers (import containers are delivered by train and export containers are delivered by truck and leave the terminal by train).

We further observed that yard manager mostly make decisions intuitively and can hardly describe their decision process or their motives. This is mainly due to the existing complexity and interdependencies in terminal operations. In fact, which policy to choose depends on the terminal configuration and applied operation strategies and workload (see table 1).

Table 1: Existing interdependencies

Factors		
terminal con-figuration	capacity	yard, tracks, truck gate, marshalling yard, equipment
	layout	yard, tracks
terminal workload	throughput	arrival rate, fluctuations, peaks, seasonality
	Interferences	train or truck delays
	container characteristics	length, stackable/not stackable, storage or transshipment containers, weight categories, road semi-trailer/swap bodies
operation strategies	equipment	allocation, scheduling, routing
	yard	allocation, marshalling, segregation
	truck gate	processing, truck parking
	tracks	grouping, pulsing, priorization
	operating hours	standard, extra hours

When choosing a storage policy one has there to take account of all factors describing the terminal. For example the yard capacity (defined by storage block characteristics as number of tiers, rows and sectors) is influenced by the arrival pattern of the containers and the container characteristics. In fact when the arrival rate and the ratio of non-stackable containers increase, the storage utilization increases and the container handling time also increases.

In order to define the best policy for a specific terminal, a decision support tool is needed. We tried to solve this problem by combining simulation and neural network methodology. Our goal is to use the capabilities of neural networks to generalize from examples, to develop a decision tool that can learn without any knowledge of the system and without any procedure formulation.

3. PROPOSED SOLUTION

Various applications of neural networks for optimization problems exist. This includes for instance transportation problems as Travelling Salesman Problems (Xu and Tsai 1990) or Shortest Path Algorithm (Zhang and Thomopoulos 1989; Soylu et al. 2000); scheduling problems (Vaithyanathan and Ignizio 1992; Johnston and Adorf 1992; Sabuncuoglu and Gurgun 1996.), combinatorial optimization problems (Lee and Sheu 1990; Sun and Nemati 2003) or dispatching problems (Vukadinovic et al. 1997; Ball 1996). We want to use neural networks to select an allocation procedure from a set of available techniques, by integrating it into a generic discrete event simulation. This approach can be classified as pattern prediction problem (Juhasz et al. 2003; Lazar and Pastravanu 2002)

The combination of simulation and neural networks is for several reasons beneficial. First the simulation of the terminal operations delivers a starting point for understanding the underlying processes. By modelling the different operations areas, one implicitly takes into account a great deal of the existing interdependencies. Further, the simulation can be used to deliver the necessary training data for the neural network. By changing step by step the simulation parameter, one can produce a great amount of example data. Finally integrating the output of the neural network into the simulation can help evaluating the performance of the neural network and can give more insight into its impacts on different terminal areas. Moreover, the integration of the neural network into the simulation can be used to dynamically define the best policy for the ongoing system status.

As system parameters change (for example a change in the arrival rate or in the transport mode priorization), the performance of the chosen storage policy may change, which makes it necessary to determine repeatedly the best suitable policy. Determining the application range of a specific storage policy is therefore an additional goal of a research.

3.1. Terminal simulation

The information needed to model the terminal system was gathered from an in-depth literature review and from on-field research of major Austrian hinterland terminals. Our goal was to develop a generic simulation model that can be used to reproduce any terminal configuration (Gronalt, Benna, and Posset 2006). For this purpose we implemented a configuration tool that can be used as a standardized questionnaire to obtain detailed information necessary to describe a terminal which is collected with the configuration tool and which can be grouped into three categories: equipment, layout and workload. As the modelling needs to be of great detail, the input data defined by the configurator is extensive. An elaborated description of all input parameters is given in Gronalt, Posset, and Benna (2007).

The defined parameters of the configuration are systematically transmitted to the simulation, where detailed lists of import and export containers are produced and edited. The goal of this data generation step is therefore to provide a quick procedure for generating detailed experiment data in the desired composition and quantity. This generation approach enables the computation of container, train and truck data in accordance with the parameter settings and thus in regard to the distributions and patterns as entered in the configuration. Further it takes into account the existing interdependencies within container terminal operations and especially among the properties of container and transport mode. For example, while allocating a specific container to a pick-up truck, container attributes and especially storage time have to be matched with the arrival time of the truck.

The standard terminal processes were then complemented by different terminal policies and all relevant activities were modelled in detail. The model was implemented in a discrete event simulation environment and includes different objects representing train, trucks, containers, equipment and storage blocks. All these objects interact by mean of predefined dynamic rules. These dynamic rules are defined as feasibility constraint, availability constraint and priority based selection rules. The feasibility rule outlines that the allocation of equipment or storage space has to be consistent with the defined layout and access possibilities for the equipment. Some combination of yard blocks and tracks or equipment may be forbidden and have to be therefore excluded in the planning phase. The availability constraint ensures that when the equipment is requested it has to be idle, which means that it is not busy, reserved or that it is not failed. Finally the priority rules ensure the constant flow of containers in the terminal as export containers a granted a higher priority than import containers. Further the service level of trucks, measured by their dwell time in the terminal, tends to be more critical and therefore trucks are served with higher priority.

For the different storage policies (as described above) we implemented different procedures. Due to the manageable size- in terms of space and throughput- of most hinterland terminals, optimization techniques are rarely used in daily operations. Instead, work is mostly done intuitively, based on decisions defined upon individual experience. To reflect the human factor of the decision process, decision rules had to be formulated. This was an important step in order to formalize the differences between existing storage policies.

The developed simulation can be used to analyse the behaviour and to evaluate the performance of different terminal configurations. By varying model parameters, the simulation can also be used to collect a large data set which can be used to train the neural network.

3.2. Approximation of target function

The target function describes the link between terminal parameters and terminal performance. To define which parameters are relevant, a set of parameter varying replications were simulated and according to the results of the scenario-analyses a sub set of parameters was chosen. Whereas the original set, defined by the configuration data counts for more than 200 parameters (see Gronalt, Posset, and Benna 2007), the final sub set was reduced to a total of 20 parameters. These parameters can be categorized into 3 groups according to the listing in table 1. The terminal configuration parameters describe mainly the existing capacities (shifting capacity per hour and equipment type and storage capacity in TEU) and the layout of the storage block within the terminal (average transport distance for the handling equipment). The terminal workload had to be considered more deeply. The defined parameters are here the average arrival rate per hour, the fluctuation level as a percent of the arrival rate, the container length mix, the average dwell time for storage and transshipment containers and the ratio of non-stackable containers. Finally the operation strategy parameters were defined by the available terminal operation time per day, the type of rail traffic (block train, shuttle or wagon load), average available time slot for loading and unloading of trains and truck processing type (on predefined pick-up position or variable pick-up position).

To quantify the impact of a parameter variation on the terminal performance, a set of performance indicators was defined. The chosen performance indicators can be grouped into 2 categories: throughput and service quality. Examples for throughput indicators are total number of moved load units and average number of served transport modes per time unit. Examples of service quality indicators are average remaining time in the system for each transport mode and ratio of unproductive moves.

Parameters and performance indicators define the input vector *v* of the neural network.

Estimating the target function is finally done by developing a radial-basis-function (rbf) neural network (Funahashi 1989), which is a special type of a two-layered Neural Network and which is particularly flexible in estimating non-linear functions. A rbf-neural networks is defined by its input neurons, one hidden layer and output neurons. Each processing unit or hidden unit of the hidden layer implement a radial activated function.

By simulating a wide range of terminal configurations and collecting the herewith generated performance indicators, the weights of the radial-basis-function can be found. The resulting data set contains the information relating configuration parameters to terminal performance and can therefore be used as a training set for the neural network, to estimate the function of terminal performance in relation to the input parameters.

3.3. Terminal optimization

Once the target function is estimated, it is integrated into a second simulation, which closely interacts with the neural network. This hybrid simulation is comparable to the first generic simulation and only differs in terms of terminal policies. In the first simulation, terminal policies are defined as simulation input at the beginning of each replication and cannot vary during the simulation run. In the second simulation (integrating the neural network), an initial policy is determined according to the target function at the beginning of the replication which can be changed afterwards if necessary. This is done with respect to changes occurring in the system, mainly due to variations in the workload. In fact, parameters are continuously monitored and sent repeatedly to the neural network which triggers a new policy when a favourable terminal performance can be expected.

The optimization problem is therefore to minimize the number of unproductive moves and to minimize the handling time per container. This is done by adjusting the storage policy in accordance with changes in terminal configuration and workload.

As a result, the integrated simulation dynamically optimizes terminal operation and alerts terminal and yard manager when a change in terminal status is occurring.

4. CONCLUSIONS

In this paper we show that the combination of simulation and neural networks techniques can be used to develop an optimization tool for hinterland terminals. Due to the complexity of terminal operations and the great number of existing parameters and interdependencies, an extensive analysis of all links relating terminal configuration and terminal performance is not possible. The explicit formulation of a non-linear function can therefore be resolved by using a neural network approximation. Especially for terminals with no decision support systems, this can be used to underline strategic decisions with regard to investments in information management systems supported by OR-techniques. In fact, terminal managers are often interested in investigated the marginal efficiency of terminal operations, which can be shown by assigning the best operating policies for a specific terminal setting.

As we describe in this paper, we concentrated on one area of the terminal, which is the storage area. In order to model all terminal processes, we still need to integrate further optimization areas. A starting point would be to consider the optimization problem which occurs when allocating and scheduling the handling equipment. For this purpose we need to develop a second neural network to estimate the relation between equipment allocation and scheduling and terminal performance. Finally the interaction of the two neural networks has to be considered.

REFERENCES

Ball, N. R., 1996. Application of a neural network based classifier system to AGV obstacle avoidance. *Mathemathics and Computers in Simulation*, 41, 285-296.

Bontekoning, Y.M., Macharis, C., Trip, J.J., 2004. Is a new applied transportation research field emerging? A review of intermodal rail-truck freight transport literature. *Transportation Research Part A*, 38, 1-34.

Funahashi, K., 1989. On the approximate realization of continous mappings by neural networks. *Neural Networks*, 2, 183-192.

Gronalt, Manfred, Benna, Thouraya, Posset, Martin, 2006. SimConT: Simulation of Hinterland Container Terminal Operations. In: Blecker, T., Kersten, W., eds. C*omplexity Management in Supply Chains- Concepts, Tools and Methods 2*, Berlin: Ericht Schmidt Verlag, 227-246.

Gronalt, Manfred., Posset, Martin, Benna, Thouraya. 2007. Standardized Configuration in the Domain of Hinterland Container Terminals. In: Blecker, T., Edwards, K., Friedrich, G., Hvam, L., Salvador, F., *eds. Series on Business Informatics and Application Systems Innovative Processes and Products for Mass Customization 3*, Berlin: GITO-Verlag, 105-120.

Johnston, M.D., Adorf, H., 1992. Scheduling with neural networks- the case of the Hubble Space Telescope. *Computers and Operations Research*, 19, 3-4, 209-240.

Juhasz, Z., Turner, S., Kunter, K., 2003. A Performance Analyser and Prediction Tool for Parallel discrete Event Simulation. *International Journal of Simulation Systems*, 4, 1-2, 7-22.

Lazar, M., Pastravanu, O., 2002. A neural predictive controller for non-linear systems. *Mathematics and Computers in Simulation*, 60, 315–324.

Lee, B., Sheu, B.J., 1990. Combinatorial optimization using competitive Hopfield Neural Networks. In: *Proceedings of the IEEE International Joint Conference on Neural Networks*, Washington, 2, 627630.

Meersmans, Patrick J.M., Dekker, Rommert, 2001. *Operations research supports container handling*. Erasmus University Rotterdam. Available from: http://www.eur.nl/WebDOC/doc/econometrie/feweco20011102151222.pdf [accessed 13.05.2006].

Rizzoli, Andrea E., Fornara, Nicoletta, Gambardella, Luca Maria, 2002. A simulation tool for combined rail/road transport in intermodal terminals. *Mathematics and Computers in Simulation*, 59, 57-71.

Sabuncuoglu, I., Gurgun, B., 1996. A neural network model for scheduling problems. *European Journal of Operational Research*, 93, 288-299.

Soylu, M., Nur, E., Kayaligil, S., 2000. A self-organizing neural network approach for the single AGV routing problem. *European Journal of Operational Research*, 121, 124-137.

Steenken, Dirk, Voß, Stefan, Stahlbock, Robert, 2004. Container terminal operation and operations research – a classification and literature review. *OR Spectrum*, 26, 3-49.

Sun, M., Nemati, H. R., 2003. Tabu Machine: A new Neural Network Solution Approach for combinatorial Optimization Problems. *Journal of Heuristics*, 9, 5-27.

Vaithyanathan, S., Ignizio, J.P., 1992. A stochastic neural network for resource constrained scheduling. *Computers and Operations Research*, 19, 3-4, 241-254.

Vis, Iris F.A., de Koster, René, 2003. Transshipment of containers at a container terminal: An overview. *European Journal of Operational Research*, 147, 1-16.

Vukadinovic, K., Teodorovic, D., Pavkovic, G., 1997. A neural network approach to the vessel dispatching problem. *European Journal of Operational Research*, 102, 473-487.

Xu, X., Tsai, W.T., 1990. An adaptive neural network algorithm for the travelling salesman problem. In: *Proceedings of the IEEE International Joint Conference on Neural Networks*, Washington, 2, 716719.

Zhang, L., Thomopoulos, S.C.A., 1989. Neural network implementation of shortest path algorithm for traffic routing in communication networks. In: *Proceedings of the IEEE International Joint Conference on Neural Networks*, Washington, 2, 591.

QUADRATIC COMPUTATIONAL CIRCUITS FOR VLSI DESIGNS

Cosmin Popa

Faculty of Electronics and Telecommunications, University Politehnica of Bucharest

cosmin_popa@yahoo.com

ABSTRACT

There will be presented two important classes of computational circuits, implementing squaring and square-rooting functions. Representing a basis for obtaining all the continuous functions, the previous mentioned functions will be implemented using MOS active devices biased in the saturation region, having the advantage of an important increasing of the circuits' frequency response. For the same reason, a current-mode operation of the circuits will be imposed, the increasing of the computing speed being associated with the independence of the circuits' performance on the technological parameters. Additionally, the VLSI implementation assures a theoretically important speed increasing of the designed circuits.
The circuits' complexity will be strongly reduced by replacing classical MOS transistors by FGMOSTs (Floating Gate MOS Transistors), while the proposed design techniques assure a relatively small dependence of the circuits' performances on the second-order effects that affect the MOS transistor operation.

Keywords: computational circuits, current-mode operation, VLSI design

1. INTRODUCTION

Computational circuits are important building blocks in telecommunications or medical equipments, finding also many applications in analog signal processing because of an important decreasing of the computational time for the VLSI implementations of mathematical functions with respect to other possible approaches. Additionally, an important reduction of the circuits' complexity and an improvement of their possibility of integration could be obtained by choosing the VLSI implementation. Due to the rapid development of CMOS VLSI technology, many analog signal-processing functions can be achieved by employing the square-law model of MOS transistors working in saturation. Based on this principle, several basic building blocks, such as multipliers, active resistors and transconductors have been developed.

The squaring and square-rooting functions represent very important functions that could be obtained in CMOS technology, representing a basis for implementing all the other continuous functions by using a polynomial series expansion. It is obvious the necessity of making a compromise between the circuit complexity and the approximation error.

There are many possibilities [1]-[5] of implementation squaring and square-rooting circuits using the quadratic characteristic of the MOS transistor in saturation. The main goals of this class of circuits are the silicon occupied area, the independence of the output current on the technological parameters (associated with an independence on temperature of circuit performances) and a small sensitivity to the second-order effects (bulk effect, channel length modulation and mobility degradation).

2. THE VLSI IMPLEMENTATION OF CURRENT-MODE SQUARER CIRCUITS

2.1. The first CMOS current squarer

The original implementation of a CMOS current squarer using a FGMOS transistor is presented in Figure 1.

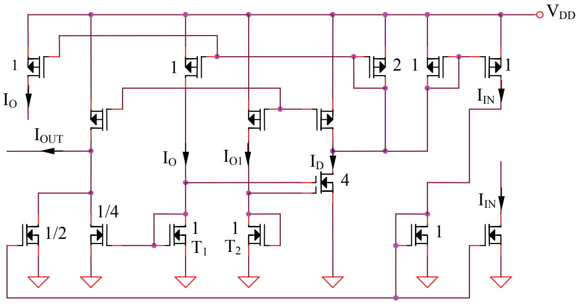


Figure 1: CMOS squarer using a FGMOS transistor

The FGMOS transistor is a MOS transistor whose gate is floating, being capacitive coupled to the multiple input gates. The drain current of a FGMOS transistor with n-input gates working in the saturation region is given by the following equation:

$$I_D = \frac{K}{2}\left[\sum_{i=1}^{n} k_i \left(V_i - V_S\right) - V_T\right]^2 , \tag{1}$$

where $K = \mu_n C_{ox}(W/L)$ is the transconductance parameter of the transistor, μ_n is the electron mobility, C_{ox} is the gate oxide capacitance, W/L is the transistor aspect ratio, $k_i, i = 1,...,n$ are the capacitive coupling ratios, V_i is the i-th input voltage, V_S is the source voltage and V_T is the threshold voltage of the transistor. Considering that all MOS transistors from Figure 1 are working in saturation and $k_1 = k_2 = 1/2$, the expression of the drain current of the FGMOS transistor could be written as:

$$I_D = \frac{4K}{2}\left(\frac{1}{2}V_{GS_1} + \frac{1}{2}V_{GS_2} - V_T\right)^2, \tag{2}$$

where V_{GS_1} and V_{GS_2} represents the gate-source voltages of T_1 and T_2 transistors, respectively. It results the following dependence of the FGMOS transistor drain current on I_O and I_{O_1} currents:

$$I_D = I_O + I_{O_1} + 2\sqrt{I_O I_{O_1}}\ , \tag{3}$$

equivalent with:

$$I_{O_1} = \frac{(I_O + I_{IN})^2}{4I_O} = \frac{I_O}{4} + \frac{I_{IN}}{2} + \frac{I_{IN}^2}{4I_O}\ . \tag{4}$$

Thus, the output current expression is $I_{OUT} = I_{IN}^2 / 4I_O$. The most important advantage of the original implementation of the current squarer proposed in Figure 1 is the absolutely (in a first-order analysis) independence of the output current on technological parameters (K, V_T).

2.2. The second CMOS current squarer

Another original idea for implementing the current-mode squarer is to use two groups of cascaded MOS transistors, the devices from the first group being biased at the same drain current, while each MOS transistor from the second group works at different drain currents.

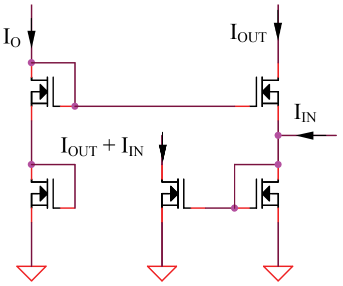


Figure 2: The current-mode CMOS squarer

Considering a strong inversion (saturation) operation of all MOS transistors from Figure 2, it is possible to write that:

$$2\sqrt{I_O} = \sqrt{I_{OUT}} + \sqrt{I_{OUT} + I_{IN}}\ , \tag{5}$$

resulting:

$$I_{OUT} = I_O - \frac{I_{IN}}{2} + \frac{I_{IN}^2}{16I_O}\ . \tag{6}$$

Subtracting I_O current and adding $I_{IN}/2$ current to the previous relation of I_{OUT}, it is possible to obtain an output current proportional to the square of the input current.

2.3. The third CMOS current squarer

The third circuit of the current squarer is presented in Figure 3.

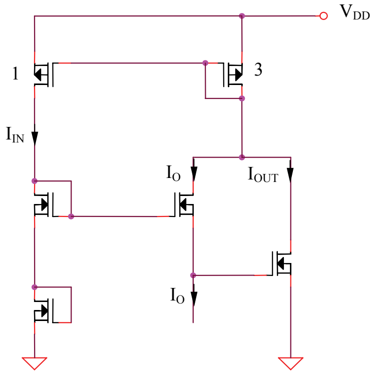


Figure 3: The circuit for implementation squaring function

Considering that all transistors from Figure 3 are working in saturation, it is possible to write:

$$2\sqrt{I_{IN}} = \sqrt{I_O} + \sqrt{I_{OUT}}\ . \tag{7}$$

Because of the PMOS current mirror having a current ratio equal to 3 it results $I_O + I_{OUT} = 3I_{IN}$. Thus, the output current of the circuit presented in the previous figure will be proportional to the square of the input current I_{IN}, $I_{OUT} = I_{IN}^2 / 4I_O$.

3. THE VLSI IMPLEMENTATION OF CURRENT-MODE SQUARE-ROOT CIRCUITS

3.1. The first square-root circuit

The new proposed implementation of the square-root circuit is based on a structure similar to the current squarer from Figure 1. The square-root circuit using MOS transistors working in saturation and a FGMOS transistor for reducing the circuit complexity is presented in Figure 4.

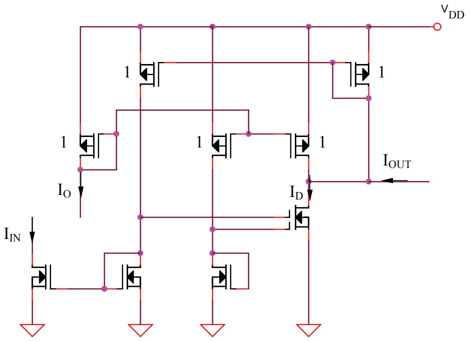


Figure 4: Square-root circuit using a FGMOS transistor

The expression of the drain current for the FGMOST from Figure 4 is:

$$I_D = I_O + I_{IN} + 2\sqrt{I_O I_{IN}} \ . \tag{8}$$

Because of the PMOS current mirrors it is possible to write $I_D = I_O + I_{IN} + I_{OUT}$, resulting the following expression of the output current:

$$I_{OUT} = 2\sqrt{I_O I_{IN}} \ . \tag{9}$$

3.2. The second square-root circuit

Considering an operation in saturation of all MOS transistors, the current I_D will have the following expression:

$$I_D = I_O - \frac{I_{OUT}}{2} + \frac{I_{OUT}^2}{16I_O} \ . \tag{10}$$

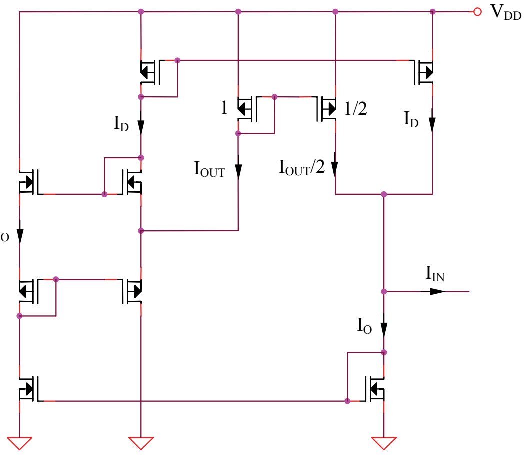


Figure 5: The square-root circuit

Because $I_D + I_{OUT}/2 = I_{IN} + I_O$, it results:

$$I_{OUT} = 4\sqrt{I_O I_{IN}} \ . \tag{11}$$

3.3. The third square-root circuit

The core of the square-root circuit is derived from the current squarer presented in Figure 2.

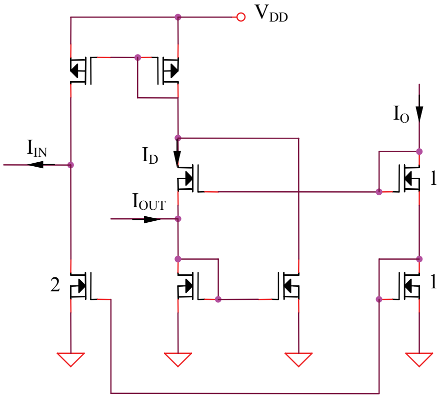


Figure 6: The current-mode square-root circuit

Similarly to the current squarer, it is possible to write that:

$$2\sqrt{I_O} = \sqrt{I_D} + \sqrt{I_D + I_{OUT}} \ , \tag{12}$$

resulting:

$$I_D = I_O - \frac{I_{OUT}}{2} + \frac{I_{OUT}^2}{16I_O} \tag{13}$$

and:

$$2\left(I_O - \frac{I_{OUT}}{2} + \frac{I_{OUT}^2}{16I_O}\right) + I_{OUT} - 2I_O = I_{IN} \ . \tag{14}$$

So, the output current I_{OUT} will be proportional with the square-root of the input current I_{IN}:

$$I_{OUT} = \sqrt{8 I_O I_{IN}} \ . \tag{15}$$

4. CONCLUSIONS

There were presented two important classes of computational circuits, implementing squaring and square-rooting functions. Representing a basis for obtaining all the continuous functions by expanding them in Taylor series, the previous mentioned functions have been implemented using exclusively MOS active devices biased in the saturation region, having the important advantage of an important increasing of the circuits' frequency response. For the same reason, a current-mode operation of the circuits has been imposed, the increasing of the computing speed being associated to the independence of the circuits' performance on the technological parameters. Additionally, the VLSI implementation of the computed functions assures a theoretically important speed increasing of the designed circuits.

The circuits' complexity has been strongly reduced by replacing classical MOS transistors by FGMOSTs (Floating Gate MOS Transistors), while the proposed design techniques assure a relatively good independence of the circuits' performances on the second-order effects that affect the MOS transistor operation.

REFERENCES

1. Popa, C., 2003. CMOS Current-Mode Pseudo-Exponential Circuits with Superior-Order Approximation. *IEEE Workshop on Nonlinear Signal and Image Processing*, pp. 25-28 (Italy).
2. Popa, C., 2003. CMOS Computational Circuits using Saturated and Bulk-Driven Weak-Inversion MOS Devices. *The 4th Electronic Circuits and Systems Conference, ECS '03*, pp. 122-125, Bratislava (Slovacia).
3. Popa, C., 2004. FGMOST-Based Temperature-Independent Euclidean Distance Circuit. *The 9th International Conference on Optimization of Electric and Electronic Equipment*, *OPTIM 2004*, pp. 29-32, Braşov (Romania).
4. Popa, C., 2004. CMOS Current-Mode Euclidean Distance Circuit Using Floating-Gate MOS Transistors. *The 24th International Conference on Microelectronics*, *MIEL 2004*, pp. 585-588.

4. Popa, C., 2004. CMOS Current-Mode High-Precision Exponential Circuit with Improved Frequency Response. *IEEE-TTTC International Conference on Automation, Quality&Testing, Robotics AQTR 2004*, pp. 279-284, Cluj-Napoca (Romania).

5. Popa, C., 2004. A Digital-Selected Current-Mode Function Generator for Analog Signal Processing Applications. *CAS 2004 Proceedings, The 27th Edition Annual Semiconductors Conference*, pp. 143-146, Sinaia (Romania).

PRACTICAL STABILITY OF POSITIVE FRACTIONAL DISCRETE-TIME LINEAR SYSTEMS

Tadeusz Kaczorek

Faculty of Electrical Engineering
Białystok Technical University
Wiejska 45D, 15-351 Białystok, Poland
kaczorek@isep.pw.edu.pl

ABSTRACT

A new concept (notion) of the practical stability of positive fractional discrete-time linear systems is introduced. Necessary and sufficient conditions for the practical stability of the positive fractional systems are established. It is shown that the positive fractional systems are practically unstable if corresponding standard positive fractional systems are asymptotically unstable.

Keywords: practical stability, fractional, positive, discrete-time

1. INTRODUCTION

In positive systems inputs, state variables and outputs take only non-negative values. Examples of positive systems are industrial processes involving chemical reactors, heat exchangers and distillation columns, storage systems, compartmental systems, water and atmospheric pollution models. A variety of models having positive linear behavior can be found in engineering, management science, economics, social sciences, biology and medicine, etc.

Positive linear systems are defined on cones and not on linear spaces. Therefore, the theory of positive systems is more complicated and less advanced. An overview of state of the art in positive systems theory is given in the monographs (Farina and Rinaldi 2000; Kaczorek 2002). Mathematical fundamentals of fractional calculus are given in the monographs (Miller and Ross 1993; Nishimoto 1984; Oldham and Spanier 1974; Podlubny 1999). The fractional positive linear continuous-time and discrete-time systems have been addressed in (Kaczorek 2008b, 2007a; Ostalczyk 2000; Vinagre and Feliu 2002; Ortigueira 1997). Stability of positive 1D and 2D systems has been addressed in (Kaczorek 2007c, 2008e, 2008f; Twardy 2007a, 2007b) and the stability of positive and fractional linear systems has been investigated in (Busłowicz 2007, 2008). The reachability and controllability to zero of positive fractional linear systems have been considered in (Kaczorek 2008a, 2007a; Klamka 2002). The fractional order controllers have been developed in (Oustalup 1993). A generalization of the Kalman filter for fractional order systems has been proposed in Sierociuk and Dzieliński 2006). Fractional polynomials and nD systems have been investigated in (Gałkowski and Kummert 2005). The notion of standard and positive 2D fractional linear systems has been introduced in (Kaczorek 2008c, 2008d).

In this paper a new concept of the practical stability of positive fractional discrete-time linear systems will be introduced and necessary and sufficient conditions for the practical stability will be established.

The paper is organized as follows.

In section 2 the basic definitions and necessary and sufficient conditions for the positivity and asymptotic stability of the linear discrete-time systems are introduced. In section 3 the positive fractional linear discrete-time systems are introduced. The main results of the paper are given in section 4, where the concept of practical stability of the positive fractional systems is proposed and necessary and sufficient conditions for the practical stability are established. Concluding remarks are given in section 5.

To the best author's knowledge the practical stability of the positive fractional systems has not been considered yet.

The following notation will be used in the paper. The set of real $n \times m$ matrices with nonnegative entries will be denoted by $R_+^{n\times m}$ and $R_+^n = R_+^{n\times 1}$. A matrix $A = [a_{ij}] \in R_+^{nxm}$ (a vector) will be called strictly positive and denoted by $A > 0$ if $a_{ij} > 0$ for $i = 1,\dots,n$, $j = 1,\dots,m$. The set of nonnegative integers will be denoted by Z_+.

2. POSITIVE 1D SYSTEMS

Consider the linear discrete-time system:

$$x_{i+1} = Ax_i + Bu_i, \quad i \in Z_+ \tag{1a}$$

$$y_i = Cx_i + Du_i \tag{1b}$$

where, $x_i \in R^n$, $u_i \in R^m$, $y_i \in R^p$ are the state, input and output vectors and, $A \in R^{nxn}$, $B \in R^{nxm}$, $C \in R^{pxn}$, $D \in R^{pxm}$.

Definition 1.
The system (1) is called (internally) positive if $x_i \in R_+^n, y_i \in R_+^p, i \in Z_+$ for any $x_0 \in R_+^n$ and every $u_i \in R_+^m, i \in Z_+$.

Theorem 1 (Farina and Rinaldi 2000, Kaczorek 2002).
The system (1) is positive if and only if

$$A \in R_+^{n\times n},\quad B \in R_+^{n\times m},\quad C \in R_+^{p\times n},\quad D \in R_+^{p\times m}. \tag{2}$$

The positive system (1) is called asymptotically stable if the solution

$$x_i = A^i x_0 \tag{3}$$

of the equation

$$x_{i+1} = Ax_i,\quad A \in R_+^{n\times n},\ i \in Z_+ \tag{4}$$

satisfies the condition

$$\lim_{i\to\infty} x_i = 0 \quad \text{for every } x_0 \in R_+^n \tag{5}$$

Theorem 2 (Farina and Rinaldi 2000, Kaczorek 2007c).
For the positive system (4) the following statements are equivalent:

1) The system is asymptotically stable
2) Eigenvalues $z_1, z_2, \ldots, z_n$ of the matrix A have moduli less 1, i.e. $|z_k| < 1$ for $k = 1,\ldots,n$
3) $\det[zI_n - A] \neq 0$ for $|z| \geq 1$
4) $\rho(A) < 1$ where $\rho(A)$ is the spectral radius defined by $\rho(A) = \max_{1\le k\le n}\{|z_k|\}$ of the matrix A
5) All coefficients $\hat{a}_i$, $i = 0,1,\ldots,n-1$ of the characteristic polynomial

$$\begin{aligned} p_{\hat{A}}(z) &= \det[I_n z - \hat{A}] = \\ &= z^n + \hat{a}_{n-1}z^{n-1} + \cdots + \hat{a}_1 z + \hat{a}_0 \end{aligned} \tag{6}$$

of the matrix $\hat{A} = A - I_n$ are positive.

6) All principal minors of the matrix

$$\bar{A} = I_n - A = \begin{bmatrix} \bar{a}_{11} & \bar{a}_{12} & \cdots & \bar{a}_{1n} \\ \bar{a}_{21} & \bar{a}_{22} & \cdots & \bar{a}_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ \bar{a}_{n1} & \bar{a}_{n2} & \cdots & \bar{a}_{nn} \end{bmatrix} \tag{7a}$$

are positive, i.e.

$$|\bar{a}_{11}| > 0, \begin{vmatrix} \bar{a}_{11} & \bar{a}_{12} \\ \bar{a}_{21} & \bar{a}_{22} \end{vmatrix} > 0, \ldots, \det \bar{A} > 0 \tag{7b}$$

7) There exists a strictly positive vector $\bar{x} > 0$ such that

$$\left[A - I_n\right]\bar{x} < 0 \tag{8}$$

Theorem 3 (Kaczorek 2002). The positive system (4) is unstable if at least one diagonal entry of the matrix A is greater than 1.

3. POSITIVE FRACTIONAL SYSTEMS

In this paper the following definition of the fractional discrete derivative

$$\Delta^\alpha x_k = \sum_{j=0}^{k} (-1)^j \binom{\alpha}{j} x_{k-j},\quad 0 < \alpha < 1 \tag{9}$$

will be used for where $\alpha \in R$ is the order of the fractional difference, and

$$\binom{\alpha}{j} = \begin{cases} 1 & \text{for } j = 0 \\ \dfrac{\alpha(\alpha-1)\cdots(\alpha-j+1)}{j!} & \text{for } j = 1,2,\ldots \end{cases} \tag{10}$$

Consider the fractional discrete linear system, described by the state-space equations

$$\Delta^\alpha x_{k+1} = Ax_k + Bu_k,\quad k \in Z_+ \tag{11a}$$

$$y_k = Cx_k + Du_k \tag{11b}$$

where $x_k \in \Re^n$, $u_k \in \Re^m$, $y_k \in \Re^p$ are the state, input and output vectors and $A \in \Re^{n\times n}$, $B \in \Re^{n\times m}$, $C \in \Re^{p\times n}$, $D \in \Re^{p\times m}$.

Using the definition (9) we may write the equations (11) in the form

$$x_{k+1}+\sum_{j=1}^{k+1}(-1)^j\binom{\alpha}{j}x_{k-j+1}=Ax_k+Bu_k,\ k\in Z_+ \qquad (12a)$$

$$y_k=Cx_k+Du_k \qquad (12b)$$

Definition 2. The system (12) is called the (internally) positive fractional system if and only if $x_k\in\Re_+^n$ and $y_k\in\Re_+^p$, $k\in Z_+$ for any initial conditions $x_0\in\Re_+^n$ and all input sequences $u_k\in\Re_+^m$, $k\in Z_+$.

Theorem 4. The solution of equation (12a) is given by

$$x_k=\Phi_k x_0+\sum_{i=0}^{k-1}\Phi_{k-i-1}Bu_i \qquad (13)$$

where Φ_k is determined by the equation

$$\Phi_{k+1}=(A+I_n\alpha)\Phi_k+\sum_{i=2}^{k+1}(-1)^{i+1}\binom{\alpha}{i}\Phi_{k-i+1} \qquad (14)$$

with $\Phi_0=I_n$.

The proof is given in (Kaczorek 2007a).

Lemma 1 (Kaczorek 2007a). If

$$0<\alpha\le 1 \qquad (15)$$

then

$$(-1)^{i+1}\binom{\alpha}{i}>0 \quad \text{for} \quad i=1,2,... \qquad (16)$$

Theorem 5 (Kaczorek 2007a). Let $0<\alpha<1$. Then the fractional system (12) is positive if and only if

$$A+I_n\alpha\in\Re_+^{n\times n},\ B\in\Re_+^{n\times m},\ C\in\Re_+^{p\times n},\ D\in\Re_+^{p\times m} \qquad (17)$$

4. PRACTICAL STABILITY

From (10) and (16) it follows that the coefficients

$$c_j=c_j(\alpha)=(-1)^j\binom{\alpha}{j+1},\quad j=1,2,... \qquad (18)$$

strongly decrease for increasing j and they are positive for $0<\alpha<1$. In practical problems it is assumed that j is bounded by some natural number h.

In this case the equation (12a) takes the form

$$x_{k+1}=A_\alpha x_k+\sum_{j=1}^{h}c_jx_{k-j}+Bu_k,\quad k\in Z_+ \qquad (19)$$

where

$$A_\alpha=A+I_n\alpha \qquad (20)$$

Note that the equations (19) and (12b) describe a linear discrete-time system with *h* delays in state.

Definition 3. The positive fractional system (12) is called practically stable if and only if the system (19), (12b) is asymptotically stable.

Defining the new state vector

$$\tilde{x}_k=\begin{bmatrix}x_k\\x_{k-1}\\\vdots\\x_{k-h}\end{bmatrix} \qquad (21)$$

we may write the equations (19) and (12b) in the form

$$\tilde{x}_{k+1}=\tilde{A}\tilde{x}_k+\tilde{B}u_k,\quad k\in Z_+ \qquad (22a)$$

$$y_k=\tilde{C}x_k+\tilde{D}u_k \qquad (22b)$$

where

$$\tilde{A}=\begin{bmatrix}A_\alpha & c_1I_n & c_2I_n & ... & c_{h-1}I_n & c_hI_n\\ I_n & 0 & 0 & ... & 0 & 0\\ 0 & I_n & 0 & ... & 0 & 0\\ ... & ... & ... & ... & ... & ...\\ 0 & 0 & 0 & ... & I_n & 0\end{bmatrix}\in\Re_+^{\tilde{n}\times\tilde{n}},\ \tilde{B}=\begin{bmatrix}B\\0\\\vdots\\0\end{bmatrix}\in\Re_+^{\tilde{n}\times m}$$

$$\tilde{C}=\begin{bmatrix}C & 0 & ... & 0\end{bmatrix}\in\Re_+^{p\times\tilde{n}},\ \tilde{D}=D\in\Re_+^{p\times m},\quad \tilde{n}=(1+h)n \qquad (22c)$$

To test the practical stability of the positive fractional system (12) the conditions of Theorem 2 can be applied to the system (22).

Theorem 6. The positive fractional system (12) is practically stable if and only if one of the following condition is satisfied

1) Eigenvalues $\tilde{z}_k$, $k=1,...,\tilde{n}$ of the matrix $\tilde{A}$ have moduli less 1, i.e.

$$|\tilde{z}_k|<1 \text{ for } k=1,...,\tilde{n} \qquad (23)$$

2) $\det[zI_{\tilde{n}} - \tilde{A}] \neq 0$ for $|z| \geq 1$

3) $\rho(\tilde{A}) < 1$ where $\rho(\tilde{A})$ is the spectral radius defined by $\rho(\tilde{A}) = \max_{1 \leq k \leq \tilde{n}} \{|\tilde{z}_k|\}$ of the matrix $\tilde{A}$

4) All coefficients $\tilde{a}_i$, $i = 0,1,\ldots,\tilde{n}-1$ of the characteristic polynomial

$$p_{\tilde{A}}(z) = \det[I_{\tilde{n}}(z+1) - \tilde{A}] = \\ = z^{\tilde{n}} + \tilde{a}_{\tilde{n}-1}z^{\tilde{n}-1} + \ldots + \tilde{a}_1 z + \tilde{a}_0 \tag{24}$$

of the matrix $[\tilde{A} - I_{\tilde{n}}]$ are positive

5) All principal minors of the matrix

$$[I_{\tilde{n}} - \tilde{A}] = \begin{bmatrix} \tilde{a}_{11} & \tilde{a}_{12} & \ldots & \tilde{a}_{1\tilde{n}} \\ \tilde{a}_{21} & \tilde{a}_{21} & \ldots & \tilde{a}_{2\tilde{n}} \\ \ldots & \ldots & \ldots & \ldots \\ \tilde{a}_{\tilde{n}1} & \tilde{a}_{\tilde{n}1} & \ldots & \tilde{a}_{\tilde{n}\tilde{n}} \end{bmatrix} \tag{25a}$$

are positive , i.e.

$$|\tilde{a}_{11}| > 0, \quad \begin{vmatrix} \tilde{a}_{11} & \tilde{a}_{12} \\ \tilde{a}_{21} & \tilde{a}_{22} \end{vmatrix} > 0, \ldots, \det[I_{\tilde{n}} - \tilde{A}] > 0 \tag{25b}$$

6) There exist strictly positive vectors $\bar{x}_i \in \Re_+^n$, $i = 0,1,\ldots,h$ satisfying

$$\bar{x}_0 < \bar{x}_1, \ \bar{x}_1 < \bar{x}_2, \ldots, \bar{x}_{h-1} < \bar{x}_h \tag{26a}$$

such that

$$A_\alpha \bar{x}_0 + c_1 \bar{x}_1 + \ldots + c_h \bar{x}_h < \bar{x}_0 \tag{26b}$$

Proof. The first five conditions 1)-5) follow immediately from the corresponding conditions of Theorem 2. Using (8) for the matrix $\tilde{A}$ we obtain

$$\begin{bmatrix} A_\alpha & c_1 I_n & c_2 I_n & \ldots & c_{h-1} I_n & c_h I_n \\ I_n & 0 & 0 & \ldots & 0 & 0 \\ 0 & I_n & 0 & \ldots & 0 & 0 \\ \ldots & \ldots & \ldots & \ldots & \ldots & \ldots \\ 0 & 0 & 0 & \ldots & I_n & 0 \end{bmatrix} \begin{bmatrix} \bar{x}_0 \\ \bar{x}_1 \\ \bar{x}_2 \\ \vdots \\ \bar{x}_{h-1} \\ \bar{x}_h \end{bmatrix} < \begin{bmatrix} \bar{x}_0 \\ \bar{x}_1 \\ \bar{x}_2 \\ \vdots \\ \bar{x}_h \end{bmatrix} \tag{27}$$

From (27) follow the conditions (26). □

Theorem 7. The positive fractional system (12) is practically stable if the sum of entries of every row of the adjoint matrix $\mathrm{Adj}[I_{\tilde{n}} - \tilde{A}]$ is strictly positive, i.e.

$$\mathrm{Adj}[I_{\tilde{n}} - \tilde{A}]^{-1} \mathbf{1}_{\tilde{n}} >> 0 \tag{28}$$

where $\mathbf{1}_{\tilde{n}} = [1 \ \ 1 \ \ \ldots \ \ 1]^T \in \Re_+^{\tilde{n}}$, T denotes the transpose.

Proof. It is well-known (Kaczorek 2007c) that if the system (22) is asymptotically stable then

$$\bar{x} = [I_{\tilde{n}} - \tilde{A}]^{-1} \mathbf{1}_{\tilde{n}} >> 0 \tag{29}$$

is its strictly positive equilibrium point for $\tilde{B}u = \mathbf{1}_{\tilde{n}}$.
Note that

$$\det[I_{\tilde{n}} - \tilde{A}] > 0 \tag{30}$$

since all eigenvalues of the matrix $[I_{\tilde{n}} - \tilde{A}]$ are positive.
The conditions (29) and (30) imply (28). □

Example 1. Check the practical stability of the positive fractional system

$$\Delta^\alpha x_{k+1} = 0.1 x_k, \quad k \in Z_+ \tag{31}$$

for $\alpha = 0.5$ and $h = 2$.

Using (18), (20) and (22c) we obtain

$$c_1 = \frac{\alpha(\alpha-1)}{2} = \frac{1}{8}, \quad c_2 = \frac{1}{16}, \quad a_\alpha = 0.6$$

and

$$\tilde{A} = \begin{bmatrix} a_\alpha & c_1 & c_2 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} = \begin{bmatrix} 0.6 & \frac{1}{8} & \frac{1}{16} \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}$$

In this case the characteristic polynomial (24) has the form

$$p_{\tilde{A}}(z) = \det[I_{\tilde{n}}(z+1) - \tilde{A}] = \begin{bmatrix} z+0.4 & -\frac{1}{8} & -\frac{1}{16} \\ -1 & z+1 & 0 \\ 0 & -1 & z+1 \end{bmatrix} = \\ = z^3 + 2.4z^2 + 1.675z + 0.2125 \tag{32}$$

All coefficients of the polynomial (32) are positive and by Theorem 6 the system is practically stable.
Using (28) we obtain

$$\mathrm{Adj}[I_{\tilde{n}} - \tilde{A}]\mathbf{1}_{\tilde{n}} = \left(\mathrm{Adj} \begin{bmatrix} 0.4 & -\frac{1}{8} & -\frac{1}{16} \\ -1 & 1 & 0 \\ 0 & -1 & 1 \end{bmatrix} \right) \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} = \begin{bmatrix} 2.0625 \\ 0.6500 \\ 1.6125 \end{bmatrix}$$

Therefore, by Theorem 7 the system is also practically stable.

Theorem 8. The positive fractional system (12) is practically stable only if the positive system

$$x_{k+1} = A_\alpha x_k, \quad k \in Z_+ \tag{33}$$

is asymptotically stable.

Proof. From (26b) we have

$$(A_\alpha - I_n)\bar{x}_0 + c_1\bar{x}_1 + \ldots + c_h\bar{x}_h < 0 \tag{34}$$

Note that the inequality (34) may be satisfied only if there exists a strictly positive vector $\bar{x}_0 \in \Re_+^n$ such that

$$(A_\alpha - I_n)\bar{x}_0 < 0 \tag{35}$$

since $c_1\bar{x}_1 + \ldots + c_h\bar{x}_h > 0$.

By Theorem 2 the condition (35) implies the asymptotic stability of the positive system (33). □

From Theorem 8 we have the following important corollary.

Corollary. The positive fractional system (12) is practically unstable for any finite *h* if the positive system (33) is asymptotically unstable.

Theorem 9. The positive fractional system (12) is practically unstable if at least one diagonal entry of the matrix A_α is greater than 1.

Proof. The proof follows immediately from Theorems 8 and 3. □

Example 2. Consider the autonomous positive fractional system described by the equation

$$\Delta^\alpha x_{k+1} = \begin{bmatrix} -0.5 & 1 \\ 2 & 0.5 \end{bmatrix} x_k, \quad k \in Z_+ \tag{36}$$

for $\alpha = 0.8$ and any finite *h*.

In this case $n = 2$ and

$$A_\alpha = A + I_n\alpha = \begin{bmatrix} 0.3 & 1 \\ 2 & 1.3 \end{bmatrix} \tag{37}$$

By Theorem 9 the positive fractional system is practically unstable for any finite *h* since the entry (2,2) of the matrix (37) is greater than 1.

The same result follows from the condition 5 of Theorem 2 since the characteristic polynomial of the matrix $A_\alpha - I_n$

$$p_{\tilde{A}}(z) = \det[I_{\tilde{n}}(z+1) - A_\alpha] = \\ = \begin{bmatrix} z+0.7 & -1 \\ -2 & z-0.3 \end{bmatrix} = z^2 + 0.4z - 2.21$$

has one negative coefficient $\hat{a}_0 = -2.21$.

5. CONCLUDING REMARKS

The new concept (notion) of the practical stability of the positive fractional discrete-time linear systems has been introduced. Necessary and sufficient conditions for the practical stability of the positive fractional systems have been established. It has been shown that the positive fractional system (12) is practically unstable for any finite *h* if the standard positive system (33) is asymptotically unstable. The considerations have been illustrated by two numerical examples.

The considerations can be easily extended for two-dimensional positive fractional linear systems. An extension of these considerations for continuous-time positive fractional linear systems is an open problem.

ACKNOWLEDGMENTS

This work was supported by Ministry of Science and Higher Education in Poland under work No NN514 1939 33 and S\WE\1\06.

REFERENCES

Busłowicz M., 2007. Robust stability of positive discrete-time linear systems with multiple delays with unity rank uncertainty structure or non-negative perturbation matrices, *Bull. Pol. Acad. Techn. Sci.* Vol. 55, No. 1, pp. 347-350.

Busłowicz M., 2008. Robust stability of convex combination of two fractional degree characteristic polynomials, *Acta Mechanica et Automatica*, (Submitted)

Farina L., Rinaldi S., 2000. *Positive Linear Systems; Theory and Applications*, J. Wiley, New York.

Gałkowski K., Kummert A., 2005. *Fractional polynomials and nD systems*. Proc IEEE Int. Symp. Circuits and Systems, ISCAS'2005, Kobe, Japan, CD-ROM.

Kaczorek T., 2002. *Positive 1D and 2D Systems*, Springer-Verlag, London.

Kaczorek T., 2007a. Reachability and controllability to zero of positive fractional discrete-time systems. *Machine Intelligence and Robotics Control*, vol. 6, no. 4. (in Press)

Kaczorek T., 2007b. Reachability and controllability to zero of cone fractional linear systems, *Archives of Control Scienes*, vol. 17, no. 3, pp. 357-367.

Kaczorek T., 2007c. Choice of the forms of Lyapunov functions for positive 2D Roesser model, *Intern. J. Applied Math. and Comp. Scienes*, vol. 17, no. 4, pp. 471-475.

Kaczorek T., 2008a. Reachability and controllability to zero tests for standard and positive fractional discrete-time systems, *Journal of Automation and System Engineering,* (in Press).

Kaczorek T., 2008b. Fractional positive continuous-time linear systems and their reachability, *Int. J. Appl. Math. Comput. Sci.*, vol. 18, no. 2 (in Press).

Kaczorek T., 2008c. Fractional 2D linear systems. *Journal of Automation, Mobile Robotics and Intelligent Systems*, vol.2, no.2.

Kaczorek T., 2008d. Positive different orders fractional 2D linear systems. *Acta Mechanica et Automatica*, (Submitted).

Kaczorek T., 2008e. Asymptotic stability of positive 1D and 2D linear systems, *Proc. of National Conference of Automation*, (in Press).

Kaczorek T., 2008f. LMI approach to stability of 2D positive systems with delays. *Multidimensional Systems and Signal Processing*, vol. 18, no. 3.

Klamka J., 2002. Positive controllability of positive systems, *Proc. of American Control Conference*, ACC-2002, Anchorage, (CD-ROM).

Miller K.S., Ross B., 1993. *An Introduction to the Fractional Calculus and Fractional Differenctial Equations*. Willey, New York.

Nishimoto K., 1984. *Fractional Calculus*. Koriama: Decartess Press.

Oldham K. B., Spanier J., 1974. *The Fractional Calculus*. New York: Academmic Press.

Ortigueira M. D., 1997. Fractional discrete-time linear systems, *Proc. of the IEE-ICASSP 97*, Munich, Germany, IEEE, New York, vol. 3, pp. 2241-2244.

Ostalczyk P., 2000. The non-integer difference of the discrete-time function and its application to the control system synthesis. *Int. J. Syst, Sci.* vol. 31, no. 12, pp. 551-1561.

Oustalup A., 1993. *Commande CRONE*. Paris, Hermés.

Podlubny I., 1999. *Fractional Differential Equations*. San Diego: Academic Press.

Sierociuk D., Dzieliński D., 2006. Fractional Kalman filter algorithm for the states, parameters and order of fractional system estimation. *Int. J. Appl. Math. Comp. Sci.*, vol. 16, no. 1, pp. 129-140.

Vinagre M., Feliu V.,2002. Modeling and control of dynamic system using fractional calculus: Application to electrochemical processes and flexible structures. *Proc. 41st IEEE Conf. Decision and Control*, Las Vegas, NV, pp. 214-239.

Twardy M., 2007. An LMI approach to checking stability of 2D positive systems, *Bull. Pol. Acad. Techn. Sci.* vol. 55,no.4, pp. 379-383.

Twardy M., 2007. On the alternative stability criteria for positive systems, *Bull. Pol. Acad. Techn. Sci.* vol. 55, no.4, pp. 385-393.

AUTHOR BIOGRAPHY

TADEUSZ KACZOREK, born on 27 April 1932 in Poland, received the MSc. PhD and DSc degrees from Electrical Engineering of Warsaw University of Technology in 1956, 1962 and 1964, respectively. In the period 1968-69 he has the dean of Electrical Engineering Faculty and in the period 1970-73 he was the prorector of Warsaw University of Technology. Since 1971 he has been professor and since 1974 full professor at Warsaw University of Technology. In 1986 he was elected a corresp. member and in 1996 full member of Polish Academy of Sciences. In the period 1988-1991 he was the director of the Research Centre of Polish Academy of Sciences in Rome. In May 2004 he was elected the honorary member of the Hungariar Academy of Sciences. He was awarded by the title doctor honoris causa by the University of Zielona Góra (2002), the Technical University of Lublin (2004), the Technical University of Szczecin (2004) and Warsaw University of Technology (2004). His research interests cover the theory of systems and the automatic control systems theory, specially, singular multidimensional systems, positive multidimensional systems and singular positive 1D and 2D systems. He has initiated the research in the field of singular 2D and positive 2D systems. He has published 21 books (six in English) and over 800 scientific papers. He supervised 65 PhD theses. He is Editor-in-Chief of Bulletin of Polish Academy of Scences, Technology Sciences and Editorial Member of about ten international journals. Tadeusz Kaczorek can be contacted at: kaczorek@isep.pw.edu.pl

ENTITIES WITH COMBINED DISCRETE-CONTINUOUS ATTRIBUTES IN DISCRETE-EVENT-DRIVEN SYSTEMS

Kristina Dammasch, Graham Horton[(a)]

[(a)]University of Magdeburg, Department of Simulation and Graphics

[(a)]kristina@sim-md.de, graham@sim-md.de

ABSTRACT

Whether a real system is approximated in a discrete or continuous simulation model depends on the more predominating type of variables determining the system's behavior over time. If both discretely and continuously changing variables have to be considered, hybrid simulation techniques have to be applied, where discrete events have an influence on the otherwise continuously changing system state. But in hybrid as well as continuous models the involved entities are usually seen as some kind of evolving fluid. If the individual entities of the system and their attributes are of particular interest this approach might be inappropriate. The paper presents a different approach to hybrid simulation based on colored stochastic Petri nets. It enables the modeling of distinguishable entities with both discrete and continuous attributes in an otherwise discrete-event driven system. An example is used to demonstrate the principles and show possible application fields.

Keywords: Discrete-Event Systems, Hybrid Simulation, Colored Stochastic Petri Nets, Psychosomatic Attributes

1. INTRODUCTION

Computer simulation is the emulation of a real or planned system in a computer program for studying the functionality and effectiveness of the system. In most simulation studies, the system's behavior over time is of particular interest. It is described by the alteration of involved variables considered as relevant for the system's behavior and thus determining the system state. These state changes are either modeled discretely, if the involved variables change at countable points in time, or continuously, if they change continuously over time. If a state change is caused by the occurrence of discrete events, the system is referred to as discrete-event system. Furthermore, it is possible that a system state is determined by both discrete and continuous variables. In that case hybrid models combine the features of discrete and continuous systems.

The description of a system also includes the involved entities. Entities in discrete-event systems are single objects that can move around within the system boundaries. They can either have no particular characteristic or be described by discrete attributes. In industrial applications where simulation is wide-spread, material and equipment involved in the production or distribution processes that has to be modeled are often represented by entities. Implemented attributes are usually simple discrete characteristics such as availability, destination in the process or the probability of defects. In continuous systems the involved entities are usually seen as some kind of evolving mass changing or moving fluidly.

But there are simulation studies requiring a more detailed model of the individual entities and all their relevant attributes. This might be the case if the entities themselves and their behavior in the system environment are of particular interest as it is often found in applications fields such as industrial psychology or health care. Furthermore, the modeling of not only several discrete but also continuously changing entity attributes might be necessary. Considering industrial psychology, continuous attributes could be psychosomatic parameters such as fatigue, alertness or performance. These can have influences on global system variables and therefore on the quality of the process and, eventually, on the overall system output. In health care applications, patients play a decisive role and, depending on the objectives of the study, their characteristics have to be taken into account for a system analysis. This might also include not only discrete parameters such as diagnosis or length of hospital stays but also continuous ones such as mood or compliance with treatment.

If these influences have to be regarded, we need a combined discrete-continuous simulation. But existing techniques only include hybrid system variables and not hybrid entity attributes. For that reason we would like to introduce a new approach to hybrid simulation models enabling the modeling of entities with both discrete and continuous attributes in an otherwise discrete-event system.

2. METHODOLOGICAL APPROACHES

Discrete-event simulations approximate systems as they evolve over time, where the system state changes instantaneously at countable points in time. These points are determined by the occurrence of discrete events that may change the state of the system. This state is described by all variables and attributes required for giving a complete image of the system at a particular time relative to the objectives of a simulation study. For each objective only the relevant and necessary details have to modeled (Banks, Carson, Nelson and Nicol 2001). The time when each type of event will occur next has to be stored in an event list. The simulation run is continued as long as there are scheduled events in the list.

Petri nets are a common paradigm for modeling such kinds of systems (Wang 1998). The complete model is represented as a graph consisting of places (states) and transitions (state changes due to occurring events) that are connected by directed arcs. The places of the net can contain any number of mobile elements of the system, referred to as tokens. These tokens are moved from place to place by the "firing" of the transition representing an event occurring in the system. So called immediate transitions fire instantaneously when becoming enabled, whereas timed transitions fire with a certain time delay. Colored Petri nets extend the concept by adding attributes to the tokens (Jensen 1997). The firing of a transition not only changes the distribution of tokens but can also modify the attributes of the tokens. With colored Petri nets, discrete-event systems can be modeled containing distinguishable mobile entities with specified attributes.

In continuous simulation models, the state of the system is changed by continuous processes, this means that the state variables change continuously over time. Typically, the dynamics of continuous state variables are modeled by differential equations specifying the rates of change of the variables over time. If these equations are simple the whole system can be solved analytically without a simulation. But for most continuous models numerical-analysis methods such as Euler or Runge-Kutta integration have to be used for solving the differential equations numerically (Law and Kelton 2000).

In hybrid simulations, combining discrete and continuous approaches, discrete events can have an effect on the otherwise continuously changing system state. Continuous or hybrid models can be used for modeling discrete-event systems if the number of involved entities is very large. In this case an aggregated mass of entities is “flowing” through the system. Fluid Petri nets are an example for such kind of modeling technique. Instead of discrete tokens one or more places can hold fluid tokens (Horton, Kulkarni, Nicol and Trivedi 1998) that may be used to approximate a large amount of discrete tokens. The main disadvantage is that entities are no longer distinguishable from each other and individual attributes and behavior are not observable.

One of the most important advantages of Petri nets in general is the graphical representation giving a clear overview of the system. Not only relations and interdependencies between involved events and elements are illustrated but also the movement of entities through the system during the simulation run. But the existing approach to a hybrid Petri net disregards individual entities which is insufficient for some kinds of applications.

Agent-based simulation considers entities with several related discrete or continuous attributes, but the concept of intelligent agents covers much more functionality than a detailed description of the entities’ characteristics (Macal and North 2005). Other functions such as decision making or machine learning often do not have to be considered when observing a discrete-event system and its behavior. As colored Petri nets capture a more appropriate amount of detail, we would like to establish a new approach for hybrid simulation based on this paradigm but focus on a more detailed modeling of the mobile entities.

3. ENTITIES WITH HYBRID ATTRIBUTES

For modeling a discrete-event system with mobile entities described by hybrid attributes we extend the existing concept of colored Petri nets by a new token definition. Instead of implementing only discrete attributes we would like to add continuous parameters. Therefore we now refer to the extended token as a hybrid token. In addition, the token has to have functions computing the continuous change of these attributes, typically, implemented by differential equations describing the rates of change over time. That way, our modeling approach combines the idea of hybrid simulation with still distinguishable entities moving around in the system.

In order to explain the idea in detail we have to take a closer look at colored Petri nets and their basic components. The most important modification is done to the colored tokens of the net. As shown in Figure 1 the new defined hybrid token contains not only discrete and continuous variables but also functions describing the changes in the values of continuous attribute. Therefore a hybrid token has its own attribute dynamics which are independent of system events. Regarding that the continuous attributes are not to be precomputed the integration steps have to be included in the global event list for proceeding the simulation run.

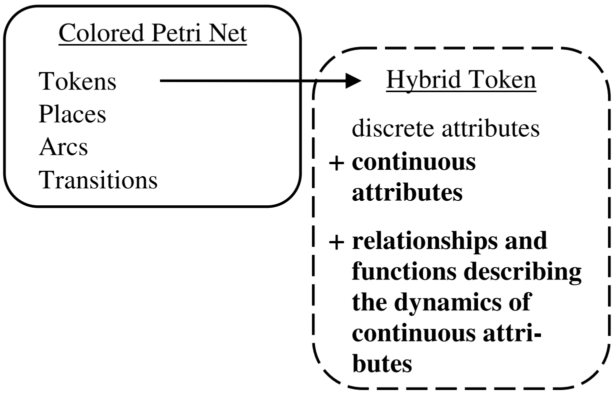


Figure 1: Components of Stochastic Petri Nets with Hybrid Tokens

Considering the hybrid token extended by continuous attributes as well as dynamics, there are several interactions possible between different token attributes. They are no longer only influenced by occurring events but by changing values of other token attributes. The following interdependencies are possible:

- A change in the value of either discrete or continuous attributes may cause a change in the value of a continuous attribute.
- A change in the value of either discrete or continuous attributes may cause the relationship governing a continuous attribute to change at particular time.
- A continuous attribute achieving a threshold value may cause a change in the value of another continuous attribute.
- A continuous attribute achieving a threshold value may cause the relationship governing another continuous attribute to change at particular time.

All other components of the Petri net remain unmodified but their functionality has to take the attribute dynamics of the hybrid token into account. Transitions can be enabled either by values of discrete or continuous token attributes and in the same manner change all attribute values as well as attribute dynamics when an event occurs.

There are also several types of possible interactions between system events and hybrid tokens. The definition follows the existing concept of combined discrete-continuous simulation (Pritsker 1995):

- A discrete event may cause a discrete change in the values of discrete or continuous token attributes.
- A discrete event may cause the relationship governing a continuous token attribute to change at particular time.
- A continuous token attribute achieving a threshold value may cause a discrete event to occur or to be scheduled.

The modeling approach of hybrid tokens adds attribute dynamics to the Petri net but keeps it separated from already existing dynamics due to the occurrence of system events. The advantages and disadvantages of the approach will be discussed using an example where human beings have to be modeled as entities in a dynamic environment.

4. MOTIVATING EXAMPLE

Assuming that our modeling approach is of particular interest for systems where human beings and their attributes have a noticeable effect on the system state, we would like to present a small illustrative example describing the behavior of human beings in their environment. First, the underlying Petri net is presented followed by details on how the idea of hybrid tokens had been implemented.

4.1. Petri Net of a Simple Queue-Server Model

The Petri net describes a simple queuing system consisting of one queue and one server illustrated in Figure 2. It is intended to model a simplified waiting behavior of customers in a queue and under which conditions a customer will leave before being served due to a loss of patience.

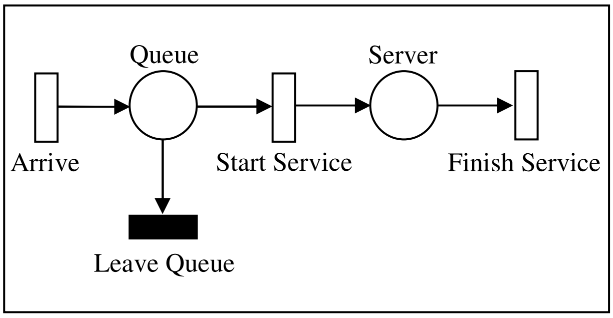


Figure 2: Petri Net for a Simple Queue-Server Model

The places “Queue” and “Server” both can contain hybrid tokens of type “Customer”. A customer in this system is characterized by the following attributes:

- patience (continuous)
- threshold (discrete)
- loyaltyToServer (discrete)
- isLeaving (discrete)

The parameter “threshold” specifies the lowest value “patience” is allowed to reach before the customer will leave the queue early. The logical attribute “isLeaving” marks this state and is set to “TRUE” whenever this case occurs. The loyalty of the customer to the server has an influence on the value of the threshold. A high loyalty (parameter is set to “TRUE”) is associated with a low threshold that will cause longer waiting times, and a low loyalty (parameter is set to “FALSE”) is associated with a high threshold that will cause shorter waiting times. Furthermore, functions need to be defined governing the evolution of “patience” over time as well as the relationship between “loyaltyToServer” and “threshold”. For the rate of change of the patience we assume a simple ordinary differential equation modeling a slow but continuous decrease of the value. An example behavior of the variable over time is shown in Figure 3.

The transition “Start Service” is enabled when “isLeaving” is “FALSE”, that is when the customer enters the queue. But as “Start Service” is a timed transition, the event marking the beginning of the service will not occur immediately but with a stochastic time delay. While time is advancing it is possible that the value for the customer’s patience will fall below the defined threshold and the attribute “isLeaving” is set to “TRUE”. This enables the transition “Leave Queue” that will fire immediately and thus remove the according token from the queue. As a consequence the scheduled event for the beginning of the service has to be removed from the event list.

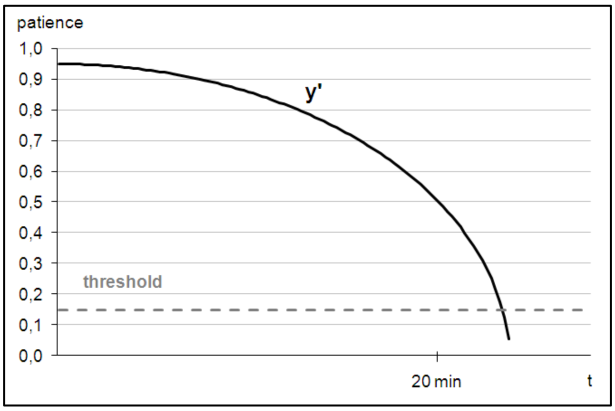


Figure 3: Example Behavior of the Continuous Variable "patience"

If the transition "Start Service" can fire, the token will be removed from the queue and added to the place "Server". After a certain amount of time the service will be finished and by the firing of "Finish Service" the token will be removed from the place "Server". The four token attributes have no influence on the timed transition.

4.2. Details on the Implementation

All attributes of the customer are initialized when the transition "Arrive" fires and creates a hybrid token. The logical attribute "isLeaving" is always initialized as "FALSE" so that the customer will never leave immediately after entering the queue. All other attributes are randomly distributed since the individual customers are to have different characteristics.

The loyalty to the server is simplified set to "TRUE" if there is some kind of loyalty, as it is observed in regular customers (Brown et al. 2003). The variable "loyaltyToServer" is set to "FALSE" if there is no particular loyalty. The ratio between being loyal and not being loyal is one to one. The initialization of this attributes also determines the threshold for the patience of the customer: In the case of loyalty the threshold is uniform distributed between 0.0 and 0.25. If the customer is disloyal the threshold is uniformly distributed between 0.26 and 0.5.

The continuous attribute "patience" is also normalized between zero and one and its initial value is assumed to be uniform distributed with a lower boundary of 0.4 and an upper boundary of 1.0. The ordinary differential equation governing the evolvement of the parameter "patience" over time is defined as

$$\frac{dy}{dt} = -c * \frac{t}{y} \qquad (1)$$

with y defined as the value of the attribute "patience". The coefficient c in the equation is also uniformly distributed with a lower boundary of 0.0002 and an upper boundary of 0.0005 influencing the amount of decrease of the customer's patience.

This differential equation describes a steady decrease over time and we choose to implement Heun's integration method that is sufficient for this simplified example. The step size used in the integration is constant and has to be defined by the user prior to the simulation run. We set the step size to 5 seconds resulting in an additional entry in the event list of the simulation model: Every 5 seconds the event "update patience" has to be executed where the integration method is executed for every hybrid token waiting in the queue.

After updating each token, a second method checks if the new value of "patience" is below the defined threshold. In that case the attribute "isLeaving" is set to "TRUE" causing the transition "Leave Queue" to fire when the update process is completed for all tokens. The step size has an influence on the accuracy and the error made when detecting the point in time when the threshold is reached.

In addition to the steady decrease of the customers' patience it is possible that the occurrence of a discrete event may cause the value of "patience" to change. An example for such an event might be the opening of a second server. The modified Petri net is illustrated in Figure 4.

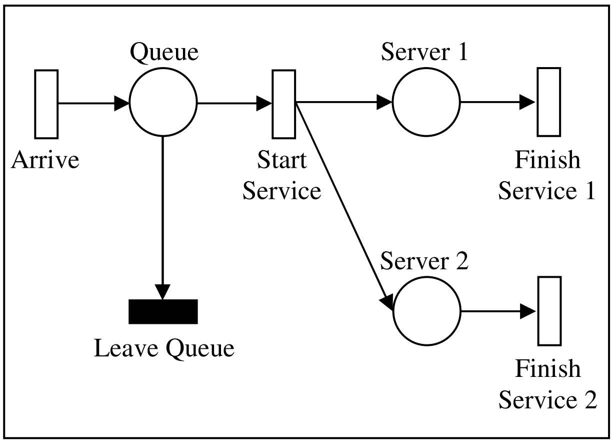


Figure 4: Petri Net with a Second Server

The continuous attribute is changed, depending on the event either immediately or scheduled, and afterwards the integration is continued based on the new value of y. Figure 5 shows a possible influence of the event on the value of "patience". The opening of the second server leads to an increase in the customers' motivation to remain in the queue.

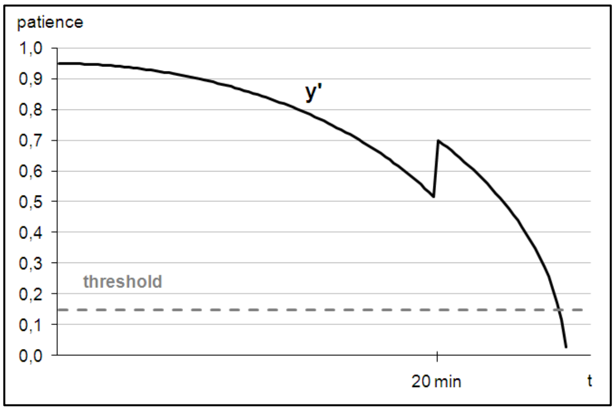


Figure 5: A Discrete Event Causes a Discrete Change in the Value of "patience"

Furthermore, it would also be possible to model the influence of a discrete event by changing the differential equation (1) governing the behavior of "patience". The opening of a second server might instead cause the parameter *c* to decrease. This would result in a slowing of the decrease of "patience". Figure 6 shows a possible development of a customer's patience after the discrete event occurred.

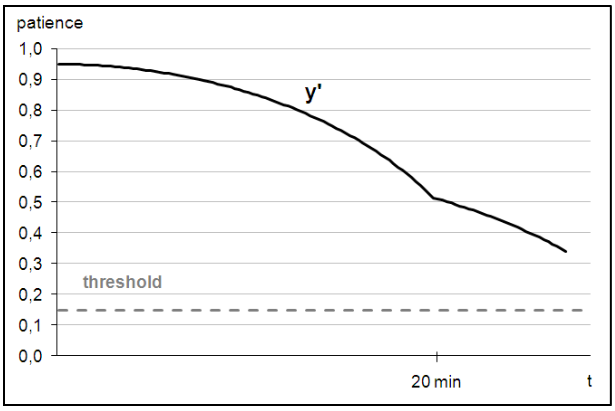


Figure 6: A Discrete Event Causes a Discrete Change in the Differential Equation

Third, it is possible that the occurrence of the discrete event causes the customers' discrete attribute "loyaltyToServer" to change. As the attribute "threshold" is depending on the loyalty it has also be updated immediately.

For our experiment we let the arrival times be exponentially distributed with rate paramter λ = 0.1 [min] and the service times normally distributed with μ = 8 [min] and σ = 5 [min]. In 500 replications around 30 percent of the customers left the queue before being served. The minimum waiting times were at 20 minutes ± 10 seconds. The maximum waiting time a customer spent in the queue was 58 minutes ± 30 seconds.

4.3. Evaluation of the Experiment

A small and simple model like this is well suited for explaining the ideas behind hybrid tokens. But at the same time the question may arise why modeling this system with the aid of hybrid and, with respect to the computational effort, "expensive" tokens. As the system only contains one continuous variable for the patience of the customer it would be easy to precompute the point in time when the customer will leave the queue early before getting served. This point in time could be scheduled in the event list and the modeling of the continuous change of the patience would not be necessary.

But when a larger number of continuous attributes are involved and the equations describing the rates of changes are not particularly simple, the preliminary computation of the values can become very complex. In that case the use of hybrid tokens might be the easier solution. Furthermore, the modeling of all necessary attributes and their dynamics directly within the token is much closer to the natural way of observing and understanding such kinds of systems.

Examples can be found in many different application fields. Considering industrial psychology the human attributes satisfaction and performance are of particular interest (Borman, Ilgen, Klimoski, and Weiner 2003). These are influenced both by several system events and by several other continuous human attributes such as mood, fatigue, or alertness. They are all mathematically related in different ways and have influences on each other as well as on the system output, for example by increasing or decreasing the performance of the human being. One example of use might be the evaluation of different strategies for increasing a worker's performance during days and night shifts.

Hybrid tokens enable the simulation of all relevant discrete and continuous attributes and their dynamics without making non-trivial pre-computations necessary. One further advantage is that the graphical representation of the Petri net is kept simple as no auxiliary transitions have to be added for being able to model all attribute dynamics.

5. SUMMARY AND CONCLUSION

Systems in practice often cannot be easily classified either as a discrete or a continuous system. But usually the system's behavior is more determined by one of the types than the other having an effect on the choice of modeling approach. If the system's design and the objectives of the simulation study require the examination of both discrete and continuous aspects hybrid simulation techniques have to be considered. But usually, these kinds of techniques do not consider distinguishable entities moving around within the system boundaries.

As we are currently developing a simulation model of the German mental health care system, we are interested in a modeling paradigm allowing the specification of entities in detail, including both discrete and continuous attributes, that are part of a discrete-event-driven system. In mental health care systems, the most important entity is the patient taking part in psychiatric services. An aggregation to a "fluid" mass may be insufficient in matters of patient-related questions, for example problems concerning mood, motivation, or the sensed quality of care. Several human attributes, especially mental ones, have to be considered making the simulation study very complex. Nevertheless, the resulting model will be used by decision makers and planners of psychiatric services who are usually not familiar with simulation techniques.

For these reasons, we introduced a new modeling approach for discrete-event-driven systems with entities that can have both discrete and continuous attributes. The approach is based on the concept of Petri nets and we refer to the entities as hybrid tokens.

This different approach to the concept of hybrid simulation may be useful for those kinds of systems where a detailed examination of involved entities with all relevant attributes is of particular interest and cannot be disregarded. The main advantage is an emphasis placed on the entities as well as their reactions and influences on the system. That might be important for all applications with human beings in dynamic environments, especially if the users of the simulation model are not simulation experts. On the one hand human characteristics can be modeled in a detailed manner but at the same time this information is encapsulated and can be hidden from the simulation user. With the aid of hybrid tokens the system and its entities can be modeled precisely and still be illustrated by a graphical representation that is easy to understand. Additionally, adding changing entity attributes directly to the entity equates more the natural way of thinking than modeling the dynamics separately from the entity.

For the above reasons, we believe that Petri nets with hybrid tokens are able to contribute to an intuitive way of modeling discrete-event systems with a close approximation of the entities involved.

REFERENCES

Banks, J., Carson, J.S., Nelson, B.L., Nicol, D.M., 2001. *Discrete-Event System Simulation*. Third Edition, Prentice-Hall, Inc.

Wank, J., 1998. *Timed Petri Nets: Theory and Application*. Kluwer Academic Publishers.

Jensen, K., 1997. *Coloured Petri Nets: Basic Concepts, Analysis Methods, and Practical Use*. Volume 1, Second Edition, Springer.

Law, A.M., Kelton, W.D., 2000. *Simulation Modeling and Analysis*. Third Edition, McGraw-Hill Higher Education.

Horton, G., Kulkarni, V.G., Nicol, D.M., Trivedi, K.S., 1998. Fluid stochastic Petri nets: Theory, applications, and solution techniques. *European Journal of Operational Research,* Volume 105 (1): Pages 184-201.

Macal, C.M., North, M.J., 2005. Tutorial on agent-based modeling and simulation. *Proceedings of the 37th Conference on Winter Simulation,* Pages 2-15. December 4-7, 2005, Orlando, FL, USA.

Brown, L., Gans, N., Mandelbaum, A., Sakov, A., Shen, H., Zeltyn, S., Zhao, L., 2002: *Statistical Analysis of a Telephone Call Center: A Queueing-Science Perspective.* Technical report. The Wharton School, University of Pennsylvania, Philadelphia.

Pritsker, A.A.B., 1995. *Introduction to Simulation and SLAM II.* Fourth Edition, John Wiley.

Borman, W.C., Ilgen, D.R., Klimoski, R.J., Weiner, I.B., 2003. *Handbook of Psychology, Volume 12: Industrial and Organizational Psychology*. Wiley & Sons, Inc.

AUTHORS BIOGRAPHY

Kristina Dammasch studied Computational Visualistics at the Otto-von-Guericke University of Magdeburg. She was awarded her German University diploma in June 2006. Since November 2006 she is working at the Research Group "Simulation and Modeling" at the University of Magdeburg. Her email address is: kristina@sim-md.de.

Graham Horton studied Computer Science at the University of Erlangen, obtaining his Masters degree ("Diplom") in 1989. He obtained his PhD in Computer Science in 1991 and his "Habilitation" in 1998 at the same university, in the field of simulation. Since 2001, he is Professor for Simulation and Modeling at the Computer Science department of the University of Magdeburg. His email address is: graham@sim-md.de.

SIMULATION MODEL OF A POLYMERIZATION PLANT

D. Gradišar[(a)], V. Jovan[(a)], S. Zorzut[(b)]

[(a)]Institute “Jožef Stefan”, Jamova 39, 1000 Ljubljana, Slovenia
[(b)]Instrumentation Technologies, Velika pot 22, 5250 Solkan, Slovenia

[(a)]dejan.gradisar@ijs.si, vladimir.jovan@ijs.si, [(b)]sebastjan@i-tech.si

ABSTRACT

The control systems in production plants are structured hierarchically into several layers, each operating on a different time scale: business-management, production and process level. In this article simulation model of the polymerization plant was presented to help developing a production control. The model was designed in Matlab, Simulink and Stateflow simulation environment. With the model it is possible to simulate the execution of scheduled jobs in production and to investigate and verify the plant wide control algorithms. In the model the process of retrieving the production Performance Indicators (pPI) is included, which are used to obtain information about current status of the production process. These indicators are also used to control the production process. Hierarchical closed-loop control scheme was proposed. Some preliminary results demonstrate the usefulness of the proposed methodology.

Keywords: production model, production control, performance indicators.

1. INTRODUCTION

The modern business environment demands an instant response to customers’ needs, a short product lifecycle, minimal inventories, short lead times, the concurrent processing of different products and short delivery times, as well as compliance with different regulations, environmental constraints, safe and reliable production measures, energy and material criteria, social pressures, changes in the workforce, etc. Advanced manufacturing requires quick and accurate decisions and actions at all management levels in a factory, and a high degree of decision-making autonomy within particular business and production processes in a factory. The demand for high cost-effectiveness has turned modern industry away from the planned production concept to an order-driven one. This has entailed a new concept of management based on an online estimation of the current situation together with efficient decision-making and execution. Great importance has been placed on the interaction and coordination of all the business and production activities in a company. These demands have established the importance of a production control system, used to perform at least two essential production-management activities: the transformation of a company’s objectives into results (products) and the optimization of production.

Production is a complex process, consisting of several interconnected operations restricted by various constraints. To be able to control the production process a lot of information has to be handled. The control systems in production plants are structured hierarchically into several layers, each operating on a different time scale (business-management level, production-management level and process level control) (Anthony 1965).

In general, an appropriate model of a production process is needed in order to cope with its behavior and to build a control system. However, this behavior is often extremely complex. When the behavior is described by a mathematical model, formal methods can be used, which usually improve the understanding of systems, allow their analysis and help in implementation. Within the changing production environment the effectiveness of production modeling is, therefore, a prerequisite for the effective design and operation of manufacturing systems.

It this paper the model of the polymer-emulsion batch-production process is presented. The polymerization process consists of three main stages: preparation of raw materials, the reaction process and the product analysis. Batches are produced successively using variety of equipment. The main purpose of designing the production-process model is the capability of simulating the execution of scheduled jobs in production and of investigating and verifying the plant-wide control algorithms. The demands on the procedural model of the case-study production process have many specifics that are not easy to implement in commercially available modeling and simulation tools. To avoid this trap, Matlab, Simulink and Stateflow simulation environment were used.

A number of information-technology products have been developed to collect and process a vast amount of production data. However, the production-management-level functions are covered only partially. The problems regarding a production manager’s decision-making process that still remain are: how to extract the relevant information from a vast amount of disposable production data in order to make the correct decision; and how to design a plant-wide production-control system that is capable of maintaining near-optimal production and eliminating a production manager's/operator's subjective assessments.

Usually the most important production objectives (such as profitability, production efficiency, plant productivity, and product quality) cannot be directly measured from current production data. For this reason their translation into a set of output production-process variables, i.e., *production-performance indicators – pPI*, should be provided (Folan and Brown 2005).

Plantwide control deals with the structural decisions of the control systems, including what to control and how to pair the variables to form the control loops (Stephanopoulos and Ng 2000). Decomposition of the problem is the underlying principle, leading to the classification of the control objectives (regulation, optimization) and the partitioning of the process for the practical implementation of the control structures. Hierarchical feedback implementation of a control is used here, where *optimization layer* computes set-points for the controlled variables and *control layer* implements this in practice, with the aim of achieving that (Larsson and Skogestad 2000).

In the next section production process and its model are described. In section 3 the development of control system is described. This building and testing of control is done on a model. Finally, some conclusions are given in section 4.

2. PRODUCTION PROCESS MODEL

2.1. Emulsion polymerization production process

The polymer-emulsion batch-production process is a typical representative of process-oriented production. The production effectiveness, to a large extent, relies on the quality of the production-control system. The production layout consists of several reactors, dosing vessels, storage tanks and equalizers, which are used for the production of different products.

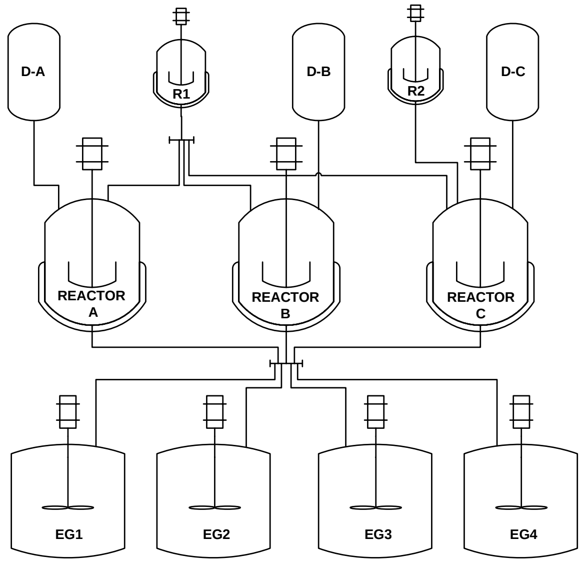


Figure 1: Simplified technological scheme of the polymerization plant

Simplified technological scheme of the polymerization plant is presented in the Figure 1. The technological process is defined with a recipe, i.e., the sequence of operations that have to be performed for the production of a particular product. Various recipes performed simultaneously can share some common resources. The polymerization process for the production of one batch of emulsion can be represented by the state-transition diagram that is depicted in Figure 2 and consists of three main stages: (i) the preparation of raw materials, (ii) the reaction process and (iii) the product analysis and reactor discharge. The optional stage of the product equalization takes place in the equalizator.

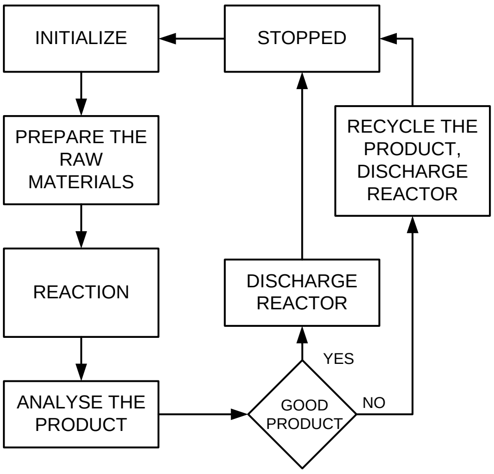


Figure 2: State-transition diagram of the polymerization process

The polymerization plant produces three groups of products (*H*, *KM* and *PA*). There are four different types of products in a product group *H* (*Homopolymers*). They follow the same process but have some differences in proportions of the raw materials and are produced in reactors A, B or C.

First input material is prepared in dosing tanks (D-A, D-B or D-C) and in reactor R1 or R2. Then the initiation starts. The reactions take place in the reactors A, B or C. After the reaction is finished more batches are mixed together in equalizer in order to equalize their properties and to obtain similar quality of a final product.

Raw materials for the second group of products (*Copolymer KM*) are prepared in dosing tank D-B and reactors R1 and R2. In this case the reaction can be performed only in reactor B.

At the end again equalization of more batches is done. Product *PA* (*Polyacrylate*) is produced in reactor E, but since this part of the process is independent of other one, it is not included in the production process model. Before the successive batch can be produced reactors have to be cleaned.

The main characteristic of this batch-production process is the production of successive batches using a variety of equipment in which intermediate products appear during each batch stage and must be used in successive stages as soon as possible. In each step certain physical actions (heating, blending) and chemical reactions are involved. The utilization of the whole production process depends on the execution of a list of production jobs.

There are many variables that are controlled during the polymerization process at the lowest (process) level of control: temperature of the reaction, reflux temperature, flow rate of monomers, pressure in the reactor, etc. Our point of interest is the control at the production level, where scheduling, plant-wide optimization etc. are performed. The variables to control here can be the quality of products, productivity, production costs…

2.2. Production process model of a polymerization plant

The demands on the model for the case-study production process have many specifics that are not easy to implement in commercially available modeling and simulation tools. To avoid this trap, the model was designed in an academically well-established Matlab, Simulink and Stateflow simulation environment.

The developed production process model of the polymerization plant represents the production process and its attributes (utilization of resources, production gain, product quality, production costs, etc) in the form needed for production management. This means that we have modeled physical realities of the process as well as production costs and quality aspects of the process. The model is structured in six logical units that are interconnected as depicted in Figure 3.

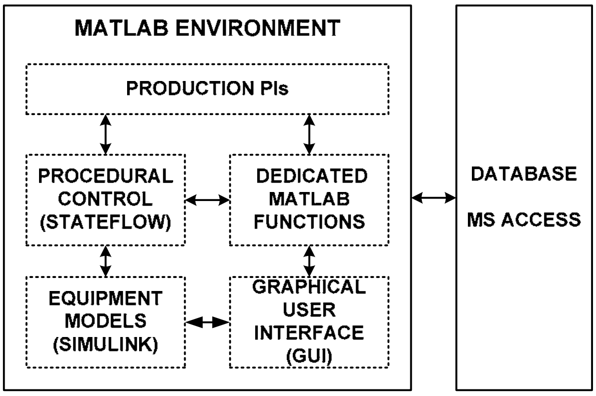

MATLAB ENVIRONMENT
PRODUCTION PIs
PROCEDURAL CONTROL (STATEFLOW)
DEDICATED MATLAB FUNCTIONS
EQUIPMENT MODELS (SIMULINK)
GRAPHICAL USER INTERFACE (GUI)
DATABASE
MS ACCESS

Figure 3: The structure of the production process model

2.3. Equipment models

The *equipment models* are created with simple Simulink models that incorporate I/O control signals. The models of the chemical reactors do not include the exact mathematical formulation of the chemical reactions involved in the polymerization process (as at this level of interest are not necessary), but they do include the equations of temperature, mass flow and level dynamics. The model of reactor R-A is demonstrated in Figure 4. On the upper part of the picture there is an integrator which describes mass flow and the lower part represents the model of a temperature.

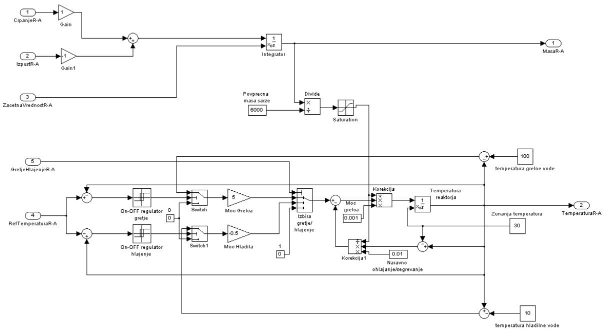


Figure 4: Simulink model of reactor R-A

2.4. Procedural control

Procedural control of the equipment was implemented using Stateflow charts. It supervises the execution of tasks and calculation of indicators about the production process status. The scheme of procedural control is depicted in Figure 5. The production jobs are scheduled according to the demands from the business management level (due times, desired product cost and quality, etc) and other production constraints (production rate, availability of resources, etc).

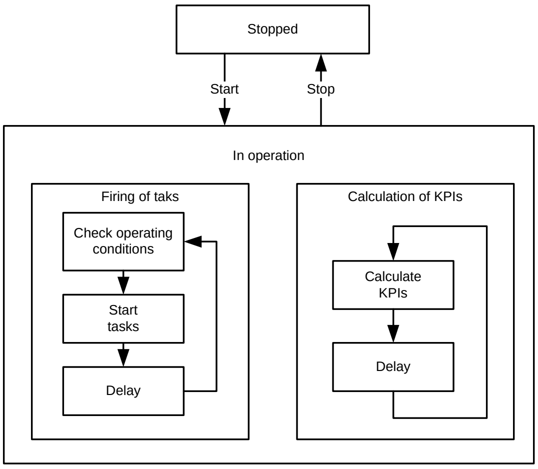


Figure 5: Procedural model scheme

2.5. Dedicated Matlab functions

Dedicated Matlab functions are used to evaluate other properties (e.g., the product quality) of the chemical reactions. These functions were designed and calibrated on the basis of statistical analyses of the production data and on knowledge about the production process obtained by interviewing production operators and technologists.

2.6. Production Performance Indicators

There are number of measured signals that can be used to get an insight into the process system state. The main problem is how to extract the relevant information from a vast amount of disposable production data in order to be used for making correct decisions in production control. The problem lies in the fact that the most important global production objectives such as profitability production efficiency, plant productivity and product quality usually cannot be directly estimated from current production data. For this reason their translation into set of output production-process variables should be provided. Production Performance Indicators (pPIs) are introduced to overcome this problem. In our case three pPIs were selected: *Productivity - P*, *Product quality - Q_P* and *Production costs - PC*. None of these pPIs is directly measurable, but an estimation of their current values can be made using the combination of the measurable output production-process variables. The procedure for the pPIs calculation has two characteristic parameters:

- calculation frequency f_{PI} which defines the time frames in which the pPIs are evaluated and
- calculation window T_{PI} that defines which production history data are used for the evaluation of the pPIs.

Productivity is defined as the amount of all products that were produced in a certain production period, and this amount is defined with:

$$P = \frac{\sum_{i=1}^{n} k_i \cdot M_i}{T} \tag{1}$$

where k_i represents the correction factor, M_i is the batch quantity, T is the observed time window and n is the number of observed batches. We take into consideration all the batches that were completely or partly produced in the defined production period and calculate the average amount of products that were produced in an hour. The correction factor defines the percentage of the production time of each batch that fits into the observed production period.

Another important indicator of production efficiency is the *mean product quality*, Q_P, which is calculated as the mean value of the quality factors of the batches that were completed in the observed production period. It is calculated with:

$$Q_P = \frac{\sum_{i=1}^{n} Q_i}{n} \tag{2}$$

where Q_i is the quality of a single batch and n is the number of observed batches.

The production costs indicator consists of *variable costs* such as raw-materials, energy, and other operating costs and *fixed costs*. The *mean production costs* (per kilogram of final product), PC, are calculated as the sum of all the costs related to production in the observed production period divided by the total number of products produced in that production period:

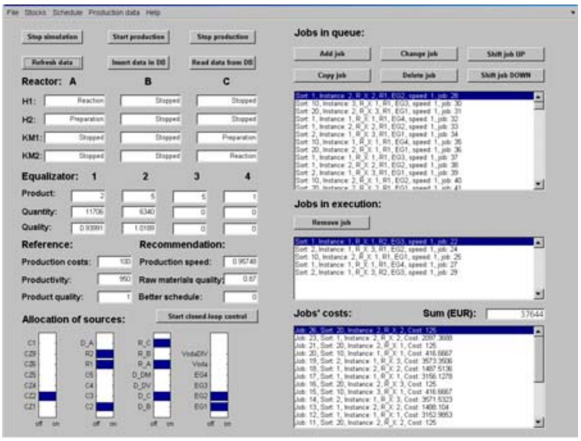
$$PC = \frac{\sum_{i=1}^{n} k_i \cdot C_i + T \cdot C_f}{\sum_{j=1}^{m} k_j \cdot M_j} \tag{3}$$

Here k_i is the correction factor for the job costs, C_i the job cost, T the production period, C_f the fixed costs, n the number of observed jobs, M_j the batch quantity, k_j a correction factor for the batch quantity and m the number of observed batches.

Those pKPIs represents the outputs of our production model. On the other hand there are more manipulated variables, i.e., physical degrees of freedom, which represent the inputs into the model. Selection of these variables is usually not much of an issue, since these variables usually follow as a direct consequence of the design of the process itself. Production jobs are scheduled according to the demands from the business management level and *Job schedule – BS* can be considered as one input variable into the production-process model. There are two other input variables that define the production process which are the *Production speed - S*, which defines the production rate, and the *Raw materials' quality - Q_{RM}*.

2.7. Graphical User Interface

The GUI (Figure 6) enables the user to simulate the production process; the user can manipulate online the *Job schedule*, the *Production speed* and the *Raw materials' quality*. On the other hand, the GUI presents the current state of the equipment (reactors, equalizator, etc.) and enables statistical analyses and a visual representation of the historical production data.



Reactor:	A	B	C
H1:	Reaction	Stopped	Stopped
H2:	Preparation	Stopped	Stopped
KM1:	Stopped	Stopped	Preparation
KM2:	Stopped	Stopped	Reaction

Equalizator:	1	2	3	4
Product:	2	5	5	1
Quantity:	11706	6340	0	0
Quality:	0.93991	1.0189	0	0

Reference:		Recommendation:	
Production costs:	130	Production speed:	0.95748
Productivity:	950	Raw materials quality:	0.87
Product quality:	1	Better schedule:	0

Jobs in queue:

Sort: 1, Instance: 2, R_X 2, R1, EG2, speed: 1, job: 28
Sort: 10, Instance: 3, R_X 1, R1, EG3, speed: 1, job: 30
Sort: 20, Instance: 2, R_X 3, R1, EG1, speed: 1, job: 31
Sort: 1, Instance: 1, R_X 1, R1, EG4, speed: 1, job: 32
Sort: 1, Instance: 2, R_X 2, R1, EG2, speed: 1, job: 33
Sort: 2, Instance: 1, R_X 3, R1, EG1, speed: 1, job: 34
Sort: 10, Instance: 1, R_X 1, R1, EG4, speed: 1, job: 35
Sort: 20, Instance: 2, R_X 1, R1, EG1, speed: 1, job: 36
Sort: 1, Instance: 1, R_X 1, R1, EG3, speed: 1, job: 37
Sort: 1, Instance: 2, R_X 2, R1, EG2, speed: 1, job: 38
Sort: 2, Instance: 1, R_X 3, R1, EG1, speed: 1, job: 39
Sort: 10, Instance: 2, R_X 1, R1, EG2, speed: 1, job: 40

Jobs in execution:

Sort: 1, Instance: 1, R_X 1, R2, EG3, speed: 1, job: 22
Sort: 2, Instance: 1, R_X 3, R1, EG2, speed: 1, job: 24
Sort: 10, Instance: 2, R_X 1, R1, EG1, speed: 1, job: 26
Sort: 1, Instance: 1, R_X 1, R1, EG4, speed: 1, job: 27
Sort: 2, Instance: 1, R_X 3, R2, EG3, speed: 1, job: 29

Jobs' costs: Sum (EUR): 37644

Job: 26, Sort: 20, Instance: 2, R_X 2, Cost: 125
Job: 23, Sort: 1, Instance: 2, R_X 2, Cost: 2097.3688
Job: 21, Sort: 20, Instance: 2, R_X 1, Cost: 125
Job: 20, Sort: 10, Instance: 1, R_X 1, Cost: 416.6667
Job: 19, Sort: 2, Instance: 1, R_X 3, Cost: 3573.3506
Job: 18, Sort: 1, Instance: 2, R_X 2, Cost: 1487.5136
Job: 17, Sort: 1, Instance: 1, R_X 1, Cost: 3156.1278
Job: 16, Sort: 20, Instance: 2, R_X 3, Cost: 125
Job: 15, Sort: 10, Instance: 3, R_X 1, Cost: 416.6667
Job: 14, Sort: 2, Instance: 1, R_X 3, Cost: 3571.5323
Job: 13, Sort: 1, Instance: 2, R_X 2, Cost: 1488.104
Job: 12, Sort: 1, Instance: 1, R_X 1, Cost: 3152.9853
Job: 11, Sort: 20, Instance: 2, R_X 2, Cost: 125

Figure 6: GUI of the simulation model

2.8. Database

The simulated data are stored in an MS Access Database and are available for online and offline processing.

3. USE OF PRODUCTION PROCESS MODEL

The model was validated by the production manager and it reflexes the actual situation in the production process. Later model was used to design and to test the production control system.

The control is being performed on different levels of decisions. The idea of hierarchical control levels is related to the so-called self-optimizing control that was presented by Skogestad (2000, 2004). Generally speaking, for every system we have available degrees of freedom (decisions), u, that we want to use in order to optimize the system operation. With the proper selection of the controlled variables, *c*, and the set-points, c_s, for these variables it is possible to operate in a near-optimal regime just by preserving these variables at defined set-points. With this approach the complex optimization problem can be translated to a simpler control problem.

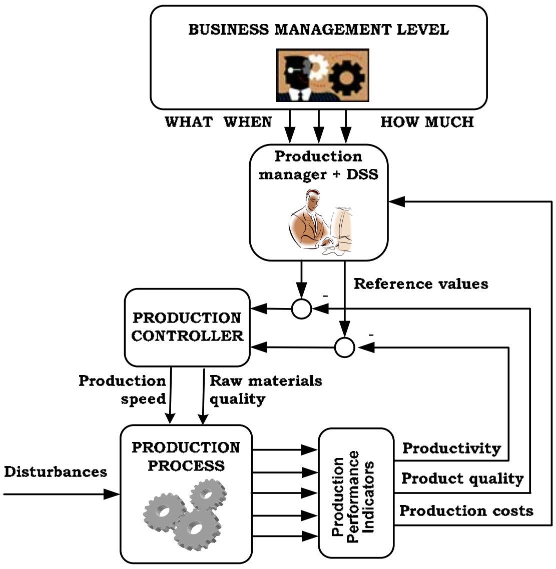


Figure 7: Hierarchical closed-loop control scheme for the polymerization process plant

Figure 7 presents the generalized, hierarchical control-loop scheme for our production process based on the self-optimizing control approach and pPIs. On the optimization level, represented by the upper control loop, the production manager optimizes the production process by selecting appropriate reference values for the pPIs in the control loop on the lower production-control level. The production manager's choice of proper pPIs set-point values depends on her/his experiences and skills, the demands from a higher business-management level and on the current state of the production process. The process of defining the set points can be improved by using the production DSS, where an estimation of the current production costs can be made using a cost model and the online production data. Once the reference values for the pPIs are defined, they are maintained by the production controller. To define appropriate controller usually also a model is needed. The described control structure reduces the complexity of the control problem; while at the upper control loop production manager with a help of a cost model make decisions about the set-point values for the chosen pPIs (e.g., on a daily basis), the lower control loop is managed automatically by the production controller more frequently (e.g., on a hourly basis).

3.1. Control of the polymerization plant

While controlling the plant, the minimization of production costs is the highest priority, and the majority of control actions are made to fulfill this demand.

On the process-optimization level the cost optimization is performed by the production manager, who is making a decisions based on the current value of the *Production costs*, the *Job schedule*, with a help of a production cost model, to define the optimal set points for the *Product quality* and *Productivity* indicators. The production costs' model acts as a kind of decision support system (DSS) for the definition of references for the pPIs.

Sensitivity analysis of the pPIs was done in order to get the production costs' model. The pPIs were evaluated at 20 working points and connected together by extrapolation. Figure 8 shows the relation between *Production costs*, *Product quality* and *Productivity* pPIs, i.e. the dependence of the *Production costs* regarding *Productivity* and *Product quality*. These are dependencies for the case for unified production, i.e., a production where a series of batches of the same or similar final products are performed on each reactor. Another cost model would be achieved for different production.

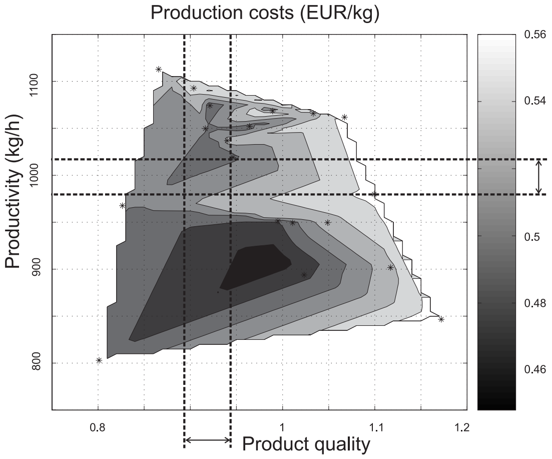


Figure 8: Cost model – *Production costs* in relation to *Productivity* and *Product quality* for unified production

Once the pPIs' reference values are defined (e.g., from the area defined with dashed lines) they are maintained by the production controller, which controls the execution of the production jobs' schedule by adjusting the available degrees of freedom for the chosen production processes, which are *Production speed* and *Raw materials' quality*.

The production controller is placed in the lower hierarchical control loop in Figure 7. To design a controller a model of the production process is needed. The part that has to be controlled is a multivariable system that can be linearized for a commonly used working area. It has two input variables (*Production speed* and *Raw materials' quality*) and two output variables (*Productivity* and *Product quality*).

Model predictive control (MPC) is well suited to solving this constraint problem (Morari and Lee 1999, Qin and Badgwell 2003). First linear process model was obtained by using the identification process over the production-process model. In the identification process, input-output data that were obtained from several simulation runs were used. Here it is assumed that the process is linear. In such a situation an approach where one input is changing while another one is fixed can be used. In the first experiment the *Raw materials' quality* was fixed and the influence of *Production speed* on the outputs of the system (*Productivity* and *Product quality*) was studied. The same experiment was repeated, but in this case the *Production speed* was fixed and the influence of *Raw materials' quality* was studied. The model parameter estimation was made using the identification method in which the least-square criterion was minimized. The input-output dependencies are therefore given with first-order models, where the *sampling time* T_S was 5 hours:

$$G = \begin{bmatrix} \dfrac{31.84}{z-0.938} & \dfrac{-4.43}{z-0.834} \\ \dfrac{-0.04}{z-0.932} & \dfrac{0.052}{z-0.94} \end{bmatrix}, \; T_S = 5\ h \qquad (4)$$

This multivariable model *G* was used for the MPC controller design, where the MPC Toolbox from the Matlab environment (Bemporad *et al.* 2006) was used.

The main challenge was the tuning of MPC controller's cost function parameters. The MPC toolbox supports the prioritizations of the outputs. In this way, the controller can provide accurate set-point tracking for the most important output, sacrificing others when necessary, e.g., when it encounters constraints. In our case the controller has to consider the input and output constraints as defined with next equations:

$$\begin{array}{l} 0.5 \le S \le 1.3 \\ 0.85 \le Q_{RM} \le 1.2 \end{array} \quad \text{and} \quad \begin{array}{l} 700 \le P \le 1300 \\ 0.87 \le Q_P \le 1.3 \end{array} \qquad (5)$$

Different weights were used to prioritize the input and output variables. To solve the optimization problem, a prediction horizon of 100 hours and a control horizon of 40 hours were used.

Closed-loop control was tested in several simulation runs. Figure 9 presents the results of an experiment where the set-point for *Productivity* was changed two times and the set-point for *Product quality* was changed just once. In the experiment a normal batch schedule for the production of three products, each of them produced in one reactor, was used. MPC managed to achieve the prescribed set-points for the controlled pPIs (*Productivity* and *Product quality*). With the increasing set-point for the *Productivity* the *Production costs* indicator is also increasing, and with the decreasing set-point for the *Product quality* the *Production costs* decrease. The *Production costs* indicator is not as smooth as the other two pPIs, which reflects the influence of the stops in production on the production costs. With an increased time horizon for the pPI evaluation such leaps in the pPI values are reduced, but also the pPI's dynamic is reduced, and consequently the performance of the MPC controller is also reduced. From the pPI responses on changed set-points for *Product quality* and *Productivity* the time constant of such a pPI model can be estimated at around 50 hours.

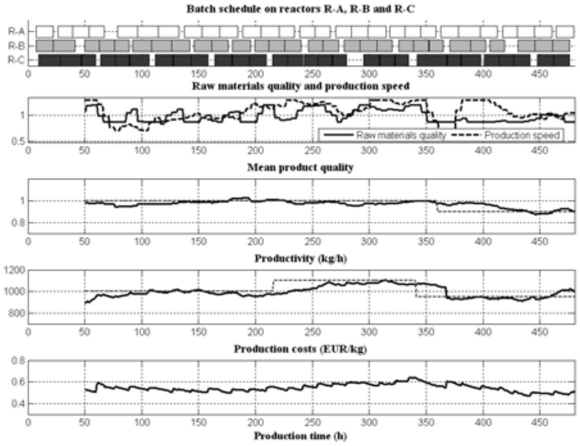


Figure 9: Batch schedule, input and output variables for one simulation run for normal production

4. CONCLUSIONS

Simulation model of a polymerization plant in the form needed for production control is presented. Production Performance Indicators are used to get an insight into the process system state. Hierarchical closed-loop control system was developed using this model and can be further implemented in the real industrial plant. The main challenge here would be the reorganization of an existing information system, in order to have available all needed variables.

REFERENCES

Anthony R., 1965. *Planning and Control Systems: A Framework for Analysis*. Division of Research, Graduate School of Business Administration, Harvard University, Boston, USA.

Bemporad, A., Morari, M., Ricker, L., 2006. *Model Predictive Control Toolbox for Use with MATLAB*, The Mathworks, Natwick.

Folan P., Brown J., 2005. A review of performance measurement: Towards performance management. *Computers in Industry*, 56, 663–680.

Larsson T., Skogestad S., 2000. Plantwide control - a review and a new design procedure. *Modeling, Identification and Control*, 21(4), 209–240.

Morari, M. and Lee, J. H., 1999. Model predictive control: Past, present and future. *Computers of Chemical Engineering*, 23 (4–5), 667–682.

Qin, S.J., Badgwell, T.A., 2003. A survey of industrial model predictive control technology. *Control Engineering Practice*, 11(7), 733–764

Skogestad, S., 2004. Near-optimal operation by self-optimizing control: from process control to marathon running and business systems. *Computers and Chemical Eng.*, 29, 127-137.

Stephanopoulos, G. and C. Ng (2000). Perspectives on the synthesis of plant-wide control structures. *J. of Process Control*, 10, 97-111.

TIMED PETRI NET SIMULATION AND RELATED SCHEDULING METHODS: A BRIEF COMPARISON

Gašper Mušič[(a)]

[(a)]University of Ljubljana
Faculty of Electrical Engineering
Tržaška 25, Ljubljana, Slovenia

[(a)]gasper.music@fe.uni-lj.si

ABSTRACT

The paper deals with Petri net based modelling and optimization of scheduling problems. Timed Petri net models are derived and used in conjunction with various optimization strategies. Standard Petri net scheduling approaches are applied, which include heuristic dispatching rules as well as heuristic based search through the reachability tree. As an alternative a simulation-based optimization is implemented to optimize the input sequences. Various conflict resolution strategies are used in order to compare and evaluate possible operation schedules in the modelled system. The strategies are based on predefined ordering rules, i.e. sequences of transition firings, which are changed during optimization procedure. The optimization problem is then solved by heuristic algorithms, including genetic algorithms, simulated annealing and threshold accepting. All these methods are implemented in the so called MATLAB PetriSimM toolbox which offers, among others, an implementation of an automated model building for the scheduling purposes. A number of benchmark tests is performed in order to compare various optimization strategies and to illustrate the suitability of the related approaches for solving practical problems.

Keywords: Petri nets, simulation, optimization, scheduling

1. INTRODUCTION

Recently, many research results related to planning and scheduling systems are reported in the engineering literature. This is due to increased demands for maximizing capacity utilization, minimizing throughput times, minimizing delays and work in progress, as well as necessity for meeting agreed delivery dates as close as possible.

Many reported results are based on formal, theoretically oriented approaches. The problems are often idealized in order to suit the constraints imposed by the chosen optimization technique and they have to ignore many practical constraints in order to solve scheduling problems efficiently. Classical mathematical programming approaches are computationally demanding and often cannot achieve feasible solutions to practical problems (Jain and Meeran 1999). This limits the applicability of classical scheduling methods in real cases found in the industrial environment.

On the other hand, simulation based scheduling is not restricted to idealized simple models. Underlying discrete-event model allows the inclusion of many process specific details and constraints. Heuristic optimization algorithms in connection with simulation systems are a suitable alternative to usual analytical methods in practical cases with high complexity (Weigert, Horn and Werner 2006). Therefore, simulation-based scheduling has potential abilities to deal with actual large-scale and complex problems (Arakawa, Fuyuki and Inoue 2002).

Advantages of discrete-event simulation include, among others, the ability to represent a system's uncertainty and dynamicity, and more generally to produce realistic (valid) representations of the real system (Semini and Fauske 2006). Nevertheless, derivation of an adequate process model may be a complex and cumbersome task. A suitable modelling framework should be chosen, allowing for a systematic model development and possible software support in model building.

Petri nets (PN) form a modelling framework that can be used through several phases of the manufacturing system life cycle (Silva and Teruel 1997). They represent a powerful graphical and mathematical modelling tool. The different abstraction levels of Petri-net models and their different interpretations make them suitable to model many aspects of manufacturing systems, including scheduling problems (Lee and DiCesare 1994, Xiong and Zhou 1998, Yu, Reyes, Cang and Lloyd 2003b). Attempts to automate the model building with Petri nets for scheduling purposes have been reported recently (Gradišar and Mušič 2007).

In the paper a Petri net modelling approach is used that is tailored to model typical scheduling problems. Various optimization strategies are then implemented using derived models. A recently proposed way of controlling the Petri net model during optimization by imposing a set of transition sequences and priorities is compared to other Petri net based scheduling approaches.

2. PETRI NETS

In the paper, Petri nets (Murata 1989, Cassandras and Lafortune 1999) are represented as Place/Transition (P/T) nets in a form of a five-tuple (P, T, I, O, M), where

- $P = \{p_1, p_2, \ldots, p_m\}, m > 0$ is a finite set of places.
- $T = \{t_1, t_2, \ldots, t_n\}, n > 0$ is a finite set of transitions (with $P \cup T \neq \emptyset$ and $P \cap T = \emptyset$).
- $I : (P \times T) \rightarrow \mathbb{N}$ is the input arc function. If there exists an arc with weight k connecting p to t, then $I(p, t) = k$, otherwise $I(p, t) = 0$.
- $O : (P \times T) \rightarrow \mathbb{N}$ is the output arc function. If there exists an arc with weight k connecting t to p, then $O(p, t) = k$, otherwise $O(p, t) = 0$.
- $M : P \rightarrow \mathbb{N}$ is the marking, M_0 is the initial marking.

Let ${}^\bullet t \subseteq P$ denote the set of places which are inputs to transition $t \in T$, i.e., there exists an arc from every $p \in {}^\bullet t$ to t. Transition t is enabled by a given marking if, and only if, $M(p) \geq I(p, t), \forall p \in {}^\bullet t$. An enabled transition can fire, and as a result removes tokens from input places and creates tokens in output places. If transition t fires, then the new marking is given by $M'(p) = M(p) + O(p, t) - I(p, t), \forall p \in P$.

In order to enable a timed analysis of the modelled system behaviour, a P/T Petri net has to be extended with time information. The concept of time is not explicitly given in the original definition of Petri nets. As described in Bowden (2000), there are three basic ways of representing time in Petri nets: firing durations, holding durations and enabling durations. The firing-duration principle says that when a transition becomes enabled it removes the tokens from input places immediately but does not create output tokens until the firing duration has elapsed. In Zuberek (1991) a well-defined description of this principle is given. When using holding-duration principle, a created token is considered unavailable for the time assigned to transition that created the token. The unavailable token can not enable a transition and therefore causes a delay in the subsequent transition firings. This principle is graphically represented in Figure 1, where the available tokens are schematized with the corresponding number of undistinguishable (black) tokens and the unavailable tokens are indicated by empty circles. The time duration of each transition is given beside the transition, e.g., $f(t_1) = t_d$. When the time duration is 0 this denotation is omitted. In Figure 1, t denotes a model time represented by a global clock and t_f denotes the firing time of a transition.

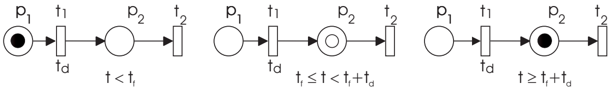


Figure 1: Timed Petri net with holding durations

With enabling durations the firing of the transitions happens immediately and the time delays are represented by forcing transitions that are enabled to stay so for a specified period of time before they can fire. When enabling duration policy is used, the firing of one transition can interrupt the enabling of other transitions, as the marking, which has enabled previous situation, has changed (Bowden 2000). It is natural to use holding durations when modelling most scheduling processes as transitions represent starting of operations, and generally once an operation starts it does not stop to allow another operation to start in between.

By using holding durations the formal representation of the timed Petri net is extended with the information of time, represented by the multiple:

$TPN = (P, T, I, O, s_0, f)$, where:

- P, T, I, O are the same as above,
- s_0 is the initial state of a timed Petri net.
- $f : T \rightarrow \mathbb{R}_0^+$ is the function that assigns a non-negative deterministic time-delay to every $t_j \in T$.

The state of a timed Petri net is a combination of three functions $s = (m, n, r)$, where,

- $m : P \rightarrow \mathbb{N}$ is a marking function of available tokens.
- $n : P \rightarrow \mathbb{N}$ is a marking function of unavailable tokens.
- r is a remaining-holding-time function that assigns values to a number of local clocks that measure the remaining time for each unavailable token (if any) in a place.

A transition t_j is enabled by a given marking if, and only if, $m(p_i) \geqslant I(p_i, t_j), \forall p_i \in \bullet t_j$. The firing of transitions is considered to be instantaneous. A new local clock is created for every newly created token and the initial value of the clock is determined by the delay of the transition that created the token. When no transition is enabled, the time of the global clock is incremented by the value of the smallest local clock. An unavailable token in a place where a local clock becomes available and the clock is destroyed. The enabling condition is then checked again.

3. PETRI NETS AND SCHEDULING

In the following, a holding durations principle of time representation in the Petri net model is assumed.

3.1. Petri net modelling of scheduling problems

An important concept in PNs is that of conflict. Two events are in conflict if either one of them can occur, but not both of them. Conflict occurs between transitions that are enabled by the same marking, where the firing of one transition disables the other transition.

The conflicts and the related conflict resolution strategy play a central role when modelling scheduling problems. This may be illustrated by a simple example, shown in Figure 2. The example involves two machines $M = \{M_1, M_2\}$, which should process two jobs $J = \{J_1, J_2\}$, and where $J_1 = \{o_1(M_1) \prec o_2(M_2)\}$ and $J_2 = \{o_3(M_1)\}$. Job J_1 therefore consist of two operations, the first one using machine M_1 and the second one machine M_2, while job J_2 involves a single operation using machine M_1. Obviously, the two jobs compete for machine M_1. This is modelled as a conflict between transitions starting corresponding operations.

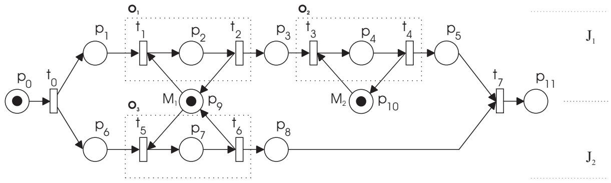


Figure 2: A PN model of a simple scheduling problem

Place p_9 is a resource place. It models the machine M_1 and is linked to t_1 and t_5, which start two distinct operations. Clearly, the conflict between t_1 and t_5 models a decision, whether machine M_1 should be allocated to job J_1 or J_2 first.

Similarly, other decisions are modelled as conflicts linked to resource places. The solution of the scheduling problem therefore maps to a conflict resolution in the given Petri net model.

3.2. Derivation of optimal or sub-optimal schedules

A derived Petri-net model can be simulated by an appropriate simulation algorithm. During the simulation, the occurring conflicts are resolved 'on the fly', e.g. by randomly choosing a transition in conflict that should fire. Instead, **heuristic dispatching rules**, such as Shortest Processing Time (SPT) or Longest Processing Time (LPT), can be introduced when solving the conflicting situations. In this way different evolutions of the Petri net can be simulated. When the marking of the places that represent resources is being considered, the schedule of process operations can be observed, i.e., when, and using which resource, a job has to be processed. Usually, different rules are needed to improve different predefined production objectives (makespan, throughput, production rates, and other temporal quantities).

To show the practical applicability of the Petri net models for the purposes of scheduling, an option for using various conflict resolution rules has been implemented within the PetriSimM tool for Matlab. With the simulation a marking trace of a timed Petri net can be achieved. The unavailable token marking trace of places that represent resources is then characterized as a schedule. In situations in which a conflict occurs, the simulator acts as a decision maker that solves the conflict. By introducing different heuristic dispatching rules (priority rules) decisions can be made easily. In this way, only one path from the reachability graph is calculated, which means that the algorithm does not require a lot of computational effort. Depending on the given scheduling problem a convenient rule should be chosen.

A more extensive exploration of the reachability tree is possible by **PN-based heuristic search method** proposed by Lee and DiCesare (1994). It is based on generating parts of the Petri net reachability tree, where the branches are weighted by the time of the corresponding operations. Sum of the weights on the path from the initial to a terminal node gives a required processing time by the chosen transition firing sequence. Such a sequence corresponds to a schedule, and by evaluating a number of sequences a (sub)optimal schedule can be determined. The method is further investigated in Yu et al. (2003), where a modified heuristic function is proposed and tested on a number of benchmark tests.

A complementary approach to the above mentioned reachability tree exploration methods is the **simulation-optimization approach**. Such a method is proposed in Löscher, Mušič and Breitenecker (2007), and is based on parametrized conflict resolution through sequences and priorities.

The main input parameters of a simulation run within the simulation-optimization approach are represented by the sequences of transition firings. A change of the input parameters leads to a different task ordering and to different schedule. Disjoint groups of transitions can be selected and to each group a firing rule can be assigned. The transitions are numbered within the group and a firing list is defined this way. All transitions of the group are deactivated except of the transition represented by the first number in the firing list. After the firing of the selected transition the next value of the list is taken.

If a priority is needed for sequence transitions a sequence priority can be defined. This sequence priority controls the behaviour of conflicts between transitions of different sequence groups. Priorities are necessary if conflicts between transitions should be solved in a special way.

When solving a scheduling problem, all possible permutations of transition sequences build the problem solution space. A fitness function is defined which assigns a value to each solution of the solution space and defines the quality of the selected solution. The evaluation of the fitness value is performed by Petri net simulation resulting in the overall cycle time of the system. The search through the solution space is driven by heuristics, like genetic algorithms, simulated annealing or threshold accepting (Vidal 1993, Dueck and Scheuer 1990, Goldberg 1989). All these methods are implemented in the so called MATLAB PetriSimM toolbox, which offers the capability of modelling, simulation, and optimisation of Timed, Coloured, and Stochastic Petri nets (Mušič, Löscher and Gradišar 2006).

4. EXAMPLES

A set of scheduling examples was chosen from the literature and tested in conjunction with the described Petri net based scheduling approaches. The examples are based on Lee and DiCesare (1994), Xiong and Zhou (1998), Yu, Reyes, Cang and Lloyd (2003b). Also two examples based on case studies were examined. These are a furniture fittings production example (Gradišar and Mušič 2007) and a production cell example (Löscher, Mušič and Breitenecker 2007).

The first example is based on semiconductor test facility studied in Xiong and Zhou (1998). The facility consists of three types of integrated circuit testers, two types of handlers and two types of hardware, which are combined into three workcenters that represent three machines in the scheduling context. There are four jobs to be scheduled, and job requirements and corresponding times are defined.

Based on the problem description, a Petri net model in Figure 3 is generated. The modelling approach follows the one suggested in Yu, Reyes, Cang and Lloyd (2003a).

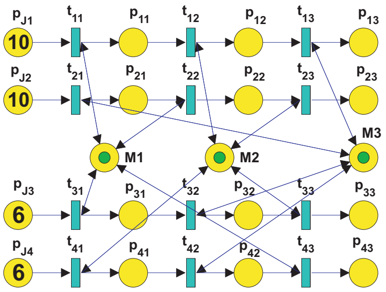


Figure 3: PN model of the first scheduling example

Places that explicitly model operation execution are omitted. Operation execution can be observed by looking for unavailable tokens in the places that follow operation triggering transitions. Time delays adjoined with these transitions model operation duration. Unavailable tokens also appear in resource places, modelling the allocation of resources.

The model has been used in conjunction with the three previously described scheduling strategies, all attempting to minimize makespan for the given lot sizes, indicated by the initial marking in the model. With the dispatching rules strategy, a so called SPT rule has been used, meaning that among any set of transitions in conflict, the one with the shortest time assignment is chosen. In case of several transitions with the same shortest time, one is randomly selected. This explains the interval 134-138 in the results presented in Table 1. Because of the occasional random selection the results of several subsequent optimization runs differ. The table also shows results of PN-heuristic search and simulations based optimization run employing simulated annealing (SA), which both result in makespan of 134. This matches the result reported in Xiong and Zhou (1998) and indicates that the given example is too simple to show any significant difference among strategies.

Next, examples from Lee and DiCesare (1994) are adopted, that have also been studied in Yu, Reyes, Cang and Lloyd (2003b). The second example (example 3 in Lee and DiCesare (1994)) deals with five jobs, each consisting of four operations in a sequence, and sharing three machines, i.e, resources (Figure 4). Lot size of 10 is assigned to each job. The main difference comparing to previous example is the flexibility of operation execution. Namely, the same operation can be executed on different machines, which also results in different execution times. This significantly increases the complexity of the optimization problem.

In order to be able to apply simulation-optimization technique, transition sequence must be defined, which will be manipulated during optimization. In the presented case this is not trivial, since the set of transitions participating in the schedule is not fixed. E.g. the first operation of job J_1 can be performed either using resource M_1 or M_3, which implies firing either t_{111} or t_{112}. The optimization should therefore not only permute the transition order in the given sequence but also change the elements of the sequence. Since this is difficult to achieve in a systematic way, the model is changed in a way which permits to apply standard combinatorial optimization techniques and a part of it is shown in Figure 5.

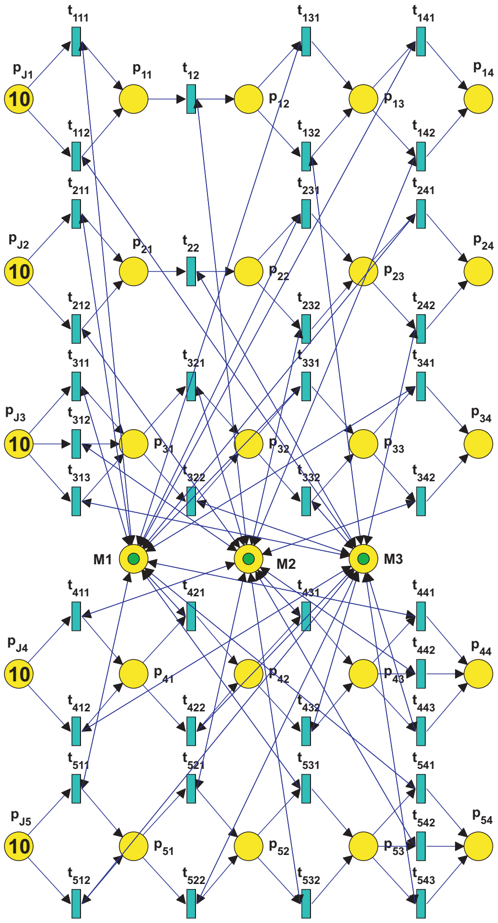


Figure 4: PN model of the second scheduling example

Table 1: Comparison of results - Examples 1-3

Example	Dispatching rules	Heuristic search	Simulation-optimization
example 1	134-138	134	134 (SA)
example 2	370-410	392	420 (SA)
example 3	254-281	258	313 (SA)

A transition-place pair is inserted before every operation triggering transition set. This newly inserted transitions always participate in the operation sequence, no matter which resource is chosen to perform the operation. A sequence of these transitions may be optimized in order to determine the best operation sequence. It must be noted, however, that such a sequence does not fully determine a possible operation schedule and therefore a certain amount of randomness must be permitted in order to cover all possible schedules. The optimization results are therefore not as good as with the other techniques, which can be observed from Table 1. The results shown in the table were obtained by simulated annealing optimization strategy, while results of threshold accepting and genetic algorithms were similar and are not shown in the table.

The third example (example 4 in Lee and DiCesare (1994)) deals with ten jobs, each with varied number of operations in a sequence, and sharing five machines and

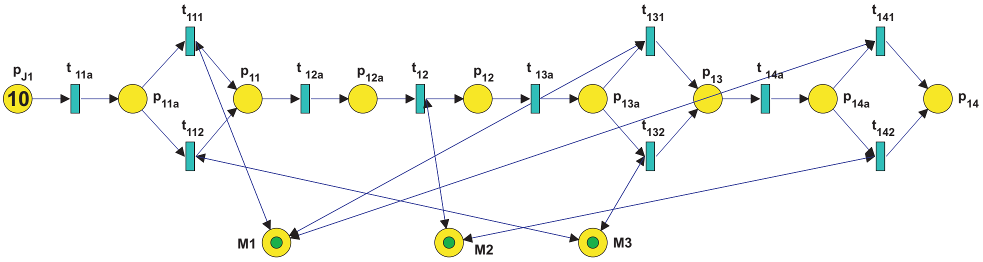


Figure 5: Part of the modified PN model of the second scheduling example

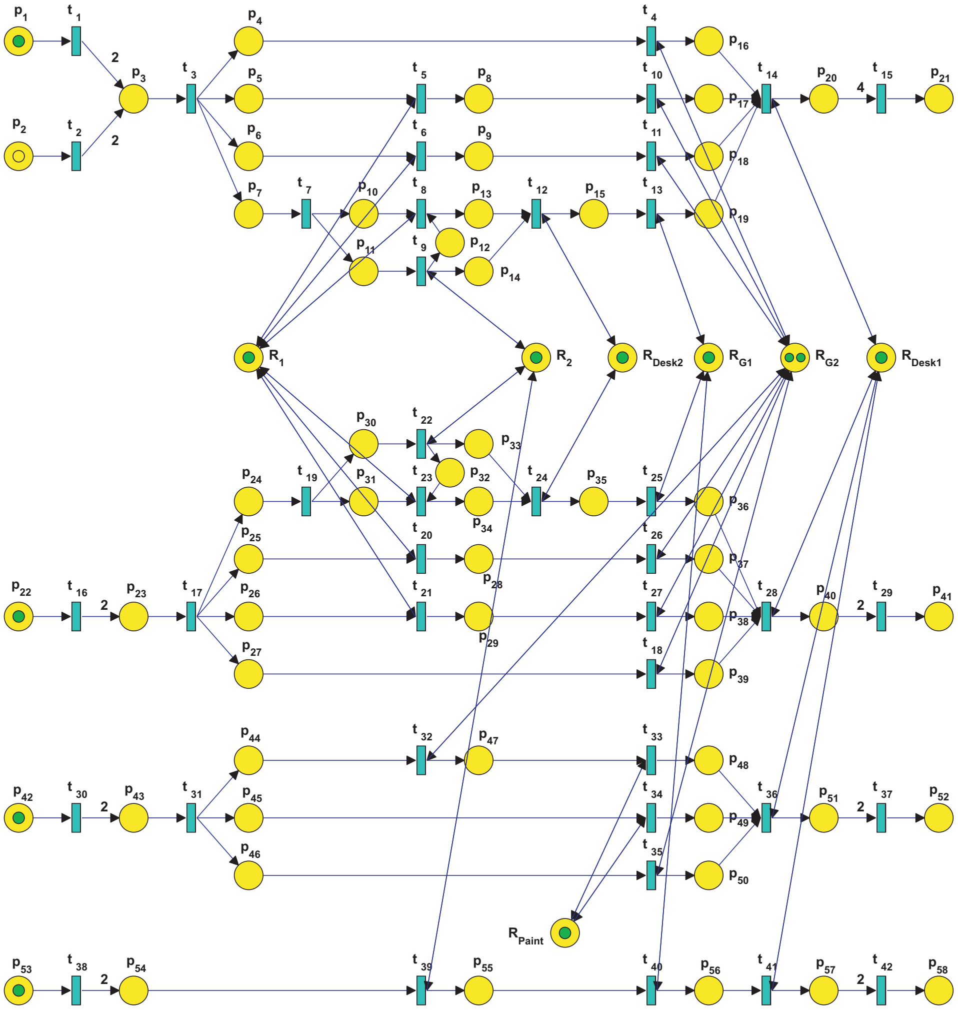


Figure 6: PN model of the furniture fittings production

three robots as resources. The example is similar to the previous one with the main difference in the varying length of jobs and corresponding lot sizes. The corresponding results are also shown in Table 1.

Finally, two case studies based examples are presented. Example 4 originates from the furniture fittings production case study presented in Gradišar and Mušič (2007). The case study deals with the production of furniture fittings described by work orders and product routing tables, and sharing resources such as assembly desks, paint and galvanization facilities etc. A model building algorithm was derived, which builds a Petri net model for a specific set of work orders. In Gradišar and Mušič (2007) such a work order set is defined and an example of the generated model is presented. An improved version of the model building algorithm is applied here, which removes redundant places and transitions and results in the simplified model shown in Figure 6.

Example 5 deals with scheduling of operations in a special type of Production cell. A number of different products is processed in the cell using special fixtures and moving platforms. The process flow in the cell consists of several steps such as setup of fixtures, setup of products, processing, dismounting of products and dismounting of fixtures. The number of platforms is limited and there is a single resource for mounting and dismounting and another

Table 2: Comparison of results - Examples 4 and 5

Example	Dispatching rules	Heuristic search	Simulation-optimization
example 4	220-230	220	220 (SA)
example 5	10160	9943	9976 (SA) 9949 (GA)

one for processing. The main scheduling problem is to determine the optimal processing order given the number of products, platforms and fixtures. The cell is described in detail in Löscher, Mušič and Breitenecker (2007).

Results of comparison for the last two examples are shown in Table 2. Again only the results obtained by simulated annealing are shown in the last column. Example 4 is rather simple from the scheduling point of view and so all techniques result in the same makespan of 220 time units. Nevertheless, the example is based on the real production in a furniture fittings production company and indicates that many real-life production problems may be adequately solved by simple dispatching rules. In such cases, the main issue is how to build appropriate model for scheduling while the quality of the optimization algorithm is of less importance.

The production cell example, on the other hand, shows the advantage of more elaborate optimization techniques. It must be noted that only the best results obtained in a number of optimization runs are shown in the table. E.g., the actual makespan interval obtained by the SPT rule was 10160-10678, while the results of the other two techniques strongly depend on parameter settings, but still gave better results in most cases.

5. CONCLUSIONS

The used methodology enables a straightforward Petri net modelling of typical scheduling problems and derivation of related schedules. It turns out, however, that not all described approaches are suitable for every type of scheduling problems. A simulation-optimization based approach, for example, is very flexible and enables incorporation of specific details in the model, but the solution of all the conflicts in the model must be parametrized in order to achieve a deterministic solution of a simulation run. This involves an extra modelling effort and is even not feasible for specific types of problems. Heuristic search based methods are easier to implement but the results are strongly dependent on the chosen heuristic function. In some cases, rather poor results are obtained if search initially starts in a non-optimal direction. The dispatching rules based methods are the easiest to implement and attractive for practice, but the results may be far from the optimum.

REFERENCES

Arakawa, M., Fuyuki, M. and Inoue, I., 2002. A simulation-based production scheduling method for minimizing the due-date deviation, *Internation Transactions in Operational Research*, 9, 153–167.

Bowden, F. D. J., 2000. A brief survey and synthesis of the roles of time in petri nets, *Mathematical & Computer Modelling*, 31, 55–68.

Cassandras, C. G. and Lafortune, S., 1999. *Introduction to Discrete Event Systems*, Kluwer Academic Publishers, Dordrecht.

Dueck, G. and Scheuer, T., 1990. Threshold Accepting: A General Purpose Optimisation Algorithm Appearing Superior to Simulated Annealing, *Journal of Computational Physics*, 90, 161–175.

Goldberg, D. E., 1989. *Genetic Algorithms in Search, Optimization, and Machine Learning*, Addision-Wesley Publishing Comp., Inc.

Gradišar, D. and Mušič, G., 2007. Production-process modelling based on production-management data: a Petri-net approach, *International Journal of Computer Integrated Manufacturing*, 20 (8), 794–810.

Jain, A. S. and Meeran, S., 1999. Deterministic job-shop scheduling: Past, present and future, *European Journal of Operational Research*, 113 (2), 390–434.

Lee, D. Y. and DiCesare, F., 1994. Scheduling flexible manufacturing systems using Petri nets and heuristic search, *IEEE Transactions on robotics and automation*, 10 (2), 123–132.

Löscher, T., Mušič, G. and Breitenecker, F., 2007. Optimisation of scheduling problems based on timed petri nets, *Proc. EUROSIM 2007*, Vol. II, Ljubljana, Slovenia.

Murata, T., 1989. Petri nets: Properties, analysis and applications, *Proc. IEEE*, 77, 541–580.

Mušič, G., Löscher, T. and Gradišar, D., 2006. An Open Petri Net Modelling and Analysis Environment in Matlab, *Proc. IMM 2006 Conf., Barcelona, Spain, 123-128*.

Semini, M. and Fauske, H., 2006. Applications of discrete-event simulation to support manufacturing logistics decision-making: A survey, *Proceedings of the 2006 Winter Simulation Conference*, pp. 1946–1953.

Silva, M. and Teruel, E., 1997. Petri nets for the design and operation of manufacturing systems, *European Journal of Control*, 3 (3), 182–199.

Vidal, R. V. V. (ed.), 1993. *Applied Simulated Annealing*, Springer-Verlag Berlin Heidelberg.

Weigert, G., Horn, S. and Werner, S., 2006. Optimization of manufacturing processes by distributed simulation, *International Journal of Production Research*, 44 (18-19), 3677–3692.

Xiong, H. H. and Zhou, M. C., 1998. Scheduling of semiconductor test facility via Petri nets and hybrid heuristic search, *IEEE Transactions on semiconductor manufacturing*, 11 (3), 384–393.

Yu, H., Reyes, A., Cang, S. and Lloyd, S., 2003a. Combined Petri net modelling and AI based heuristic hybrid search for flexible manufacturing systems-part I: Petri net modelling and heuristic search, *Computers and Industrial Engineering*, 44 (4), 527–543.

Yu, H., Reyes, A., Cang, S. and Lloyd, S., 2003b. Combined Petri net modelling and AI based heuristic hybrid search for flexible manufacturing systems-part II: Heuristic hybrid search, *Computers and Industrial Engineering*, 44 (4), 545–566.

Zuberek, W. M., 1991. Timed petri nets: definitions, properties and applications, *Microelectronics and Reliability*, 31 (4), 627–644.

ARTIFICIAL NEURAL NETWORK-BASED CLASSIFICATION OF VECTOR SETS FOR SURFACE INSPECTION

Michael Gyimesi[a], Felix Breitenecker[b], Wolfgang Heidl[c], Christian Eitzinger[d]

[a] [b]Vienna Technical University, Wiedner Hauptstr. 8-10, Wien, Austria
[c] [d]Profactor GmbH, Im Stadtgut A2, 4470 Steyr, Austria

[a]mgyimesi@osiris.tuwien.ac.at, [b]fbreiten@osiris.tuwien.ac.at, [c]Wolfgang.Heidl@profactor.at,
[d]Christian.Eitzinger@profactor.at

ABSTRACT

In applications of pattern recognition a set of objects - usually represented by feature vectors - is extracted from an image and needs to be classified as a whole set of objects, meaning that some properties of the set of objects are an aggregation of the single feature vectors – and the classification of the set of objects may depend on exactly these properties. If the number of objects, respectively the number of features, is not known or limited a priori standard classification algorithms such as support vector machines or linear classifiers cannot applied in a straightforward way due to the fixed size of the number of features in these methods. Therefore the set of object's "structure" may not be implemented properly.

In this paper we will discuss some issues of these problems and propose recurrent neural networks (RNN) as a promising method to use for such problems.

Keywords: pattern recognition, neural networks, feature aggregation, classification problem

1. INTRODUCTION

In many applications of image processing respectively pattern recognition, a set of objects - usually represented by feature vectors - is extracted from an image and needs to be classified. The classification decision can rest on properties represented by single feature vectors e.g. if the maximum 'length' of a feature vector exceeds a certain threshold. These problems are typically solved in an algorithmic way.

Another type of classification problems deals with properties, constituted by a fixed number of feature vectors. Typical examples for these problems are e.g. signal processing, data mining problems. To tackle such problems, one typically uses classification algorithms like linear classifiers, support vector machines or feedforward neural networks.

Things become more complicated, if the number of feature vectors is not known a priori.

Typical examples are

- Surface Inspection: A number of faults (= "objects") on the surface is extracted and each one is represented by a feature vector containing e.g. it's size, position, or shape. In order to achieve a proper classification e.g. good/bad, possible reasons for a "bad" decision may be the number of faults, their spatial distribution or other, even more complicated, aggregated properties. Therefore, the whole set of objects needs to be taken into consideration.
- Object Recognition: Recognizing objects that are composed of several components - e.g. an object that consists of an unspecified number of rectangles (windows), some significant edges (the walls) that provide a frame for the windows and a roof on top of all the windows can be identified as a house.
- Biomedical Imaging: The crystallization patterns of dried biological fluids, in particular native blood drops (clots) on a common medical slide, seem to contain a lot of information about diseases and other pathological disorders. Geometrical and color features are extracted out of scanned images from these clots, but only the accumulation of all these varying features of different regions of the blood spot decides, whether the patient has some pathological disorder.

The common requirements of these classification tasks:

- The number of objects in the set is different for each image.
- Aggregated information from the whole set of objects is necessary.
- The classifier has to be trainable, since the classification rules are not known beforehand.

Since our main application is surface inspection we cannot guarantee that the low level procedures produce ordered results:

- The result of the classification has to be invariant up to the order in which the objects represented by single feature vectors are aggregated.

Currently, there exist no mathematical structures that fulfill all of these properties to a high degree. In particular, the problem of trainable aggregation functions has not been investigated in great detail and most standard classification methods, such as linear classifiers, feedforward neural networks or support vector machines use a pre-specified size of feature vectors.

In the current work, these problems are not completely solved. We built different scenarios with partially simplified problems and investigated them by applying existing methods like feedforward neural networks to these problems. Furthermore, recurrent neural networks as a well suited method are one proposition to tackle these problems.

The paper is organized as follows. Section 2 explains the used methods and introduces artificial neural networks, in particular recurrent neural networks.

Section 3 provides details about the scenarios, the test data, mostly related to the application in pattern recognition.

While section 4 presents some information about the experiments and results of our work we conclude with an outlook at future work in section 5.

2. METHODS

We will investigate classification methods that take a set of feature vectors

$$X = \{x_1, x_2, ..., x_n\} \quad with\ x_i \in R^m \tag{1}$$

as input and provide a binary classification result {0,1}. The feature vectors x_i are supposed to be of the same dimension and to contain the same attributes, but there is no other relationship or structure in the data and the numbers of vectors *n* may vary for each image.

A possible solution is to arrange the feature vectors the feature vectors in a single vector,

$$\left[x_1^T, x_2^T, ..., x_n^T, 0^T, 0^T, ..., 0^T\right]^T \quad \in R^{m \cdot n_{\max}} \tag{2}$$

to fill it with zeros up to a pre-specified dimension $m \cdot n_{\max}$ and to apply a standard classification method, such as linear classifiers, feedforward neural networks, support vector machines, or any other classification method. The maximal number n_{max} of feature vectors needs to be specified and the dimension of the combined vector is n_{max} times the number of features *m*.

2.1. Linear Classifiers

Linear classifiers provide a good basis for comparison of the results as they are fully understood with respect to their properties. They try to find a hyperplane in the feature space which separates the samples into two classes (Figure 1).

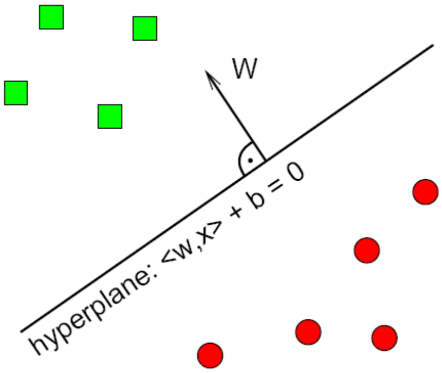


Figure 1: Linear classifier with separating hyperplane

The resulting classification problem is usually prohibitively big and requires training sets of very large size. Moreover, this trivial method does not consider the specific nature of the set of objects.

2.2. Artificial Neural Networks

A modeling method which is widely used is the classification artificial neural networks. The simplest type of artificial neural networks is a feedforward neural network. Equally structured neurons are arranged in layers which are connected by unidirectional connections. There is usually one input layer, one or more hidden layers and one output layer (Figure 2). In our application, classifying images as good or bad, only one output neuron is required to indicate the classification result.

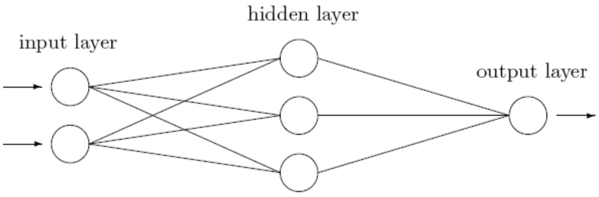


Figure 2: Feedforward neural network

While feedforward neural networks are very successful in classification and are applied in different areas like speech recognition and signal processing they have some undesirably problems. On the one hand they act as a black box and the implemented functions do not really connect to the way humans think about the original problem. On the other hand the original data has to be encoded in a finite dimensional vector space with a prior known dimension which leads to a loss of structural information.

2.3. Recurrent Neural Networks

Recurrent neural networks use, unlike simple feedforward networks, recursive neurons to build feedback structures. These neurons may have connections to all other neurons, meaning that connections in the opposite direction of the information flow will occur. This composition allows saving information on already given input data up to a certain degree and to learn regularities on structured data. Since the dynamic of recurrent neural networks are linked to the dynamics of finite automata, an interpretation of results via classical formalism is more likely than an interpretation of results of a feedforward neural network (Hammer 2000).

Furthermore, neural networks of this type can deal with a variable number of input vectors and it is not necessary to adapt the network size according to the number of data objects to process. This may be a solution to one of the key problems of the classification task –the number of objects in the set is different for each problem and not known before.

Assuming that networks with a recurrent structure require "no" limits on the number of feature vectors, they can be combined with some added feedforward network layers to generate the desired output. A typical recurrent neural network is shown in Figure 3.

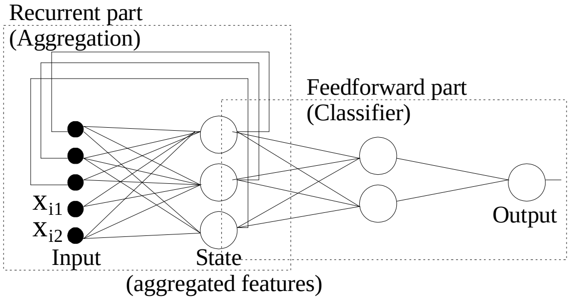


Figure 3: Simple recurrent network combined with feedforward neural network as classifier

Recurrent neural networks have been widely investigated in the field of time series classification/prediction, e.g. (Connor, Martin and Atlas 1994). Various structures have been proposed, e.g. Elman, Jordan, Wiliams-Zipser, recursive auto-associative memory (Pollack 1990) and a number of training methods have been developed; the most common seems backpropagation through time (Werbos 1990).

2.3.1. Related Work

Authors that dealt with the issue of classification of structures (Sperduti and Starita 1997; Frasconi, Gori and Sperduti 1998; Hammer 2000; Hammer 2001; Hammer 2002) propose to use different generalized recursive neurons to represent the structure of a graph. They relate their method to standard neural networks (for the classification of single patterns) and recurrent neural networks (for the classification of sequences).

Recent literature adresses other related problems such as multi-instance learning (Ramon and De Raedt 2000), long-short-term memory (Hochreiter and Schmidhuber 1997), classifying relational data (Uwents, and Blockeel 2005; Perlich and Provost 2003) and aggregation functional (Ovchinnikov and Dukhovny 2002).

3. SCENARIOS AND TEST DATA

We developed different scenarios with various test data and tasks to solve.

3.1. Comparison of different classifiers

A total of 10 test datasets, each one comprising 20.000 pictures with a different number of flaws and features per flaw but with a maximum of 24 features per picture was specified.

The specified tasks to solve where defined as follows:

3.1.1. Comp1 – Sum of a vector

The feature vector is added up and rated as "bad" if the sum fails to exceed a certain threshold value.

3.1.2. Comp2 - Areamax

If the maximal value of the area of one flaw exceeds some arbitrary number, the picture is rated as "bad".

3.1.3. Comp3 - Areasum

One feature of each flaw, which can for instance represent the area and is nonnegative, is added up for all feature vectors. The picture is rated as "bad" if the total area of flaws exceeds a certain threshold value.

3.1.4. Comp4 – Number of flaws

If the number of flaws per picture is greater than some fixed number, the picture is rated as "bad".

3.1.5. Comp5 – Minimal distance

The minimal distance between any two flaws is determined, which demands high computing effort. If it lies beneath a certain threshold, the picture is rated as "bad".

3.1.6. Comp6 – Cluster formation

A rather advanced classification task is to investigate the formation of clusters within the data points. In order to detect a cluster, the number of data points in the neighborhood of every data point was determined.

3.1.7. Comp7 – Number of flaws in a specific region

This criterion rates a picture as "bad" if the number of flaws in a certain region exceeds some arbitrary threshold value.

The according parameters where chosen in a way, that approximately half of the pictures are rated as "good" respectively "bad".

3.2. Evaluation of recurrent artificial networks

In order to evaluate the performance of different network architectures and network sizes, we have generated a total of eight data sets with varying complexity. All but one data set require utilizing aggregated information of the flaws present in one image to solve the associated classification task. One data set requiring no aggregate information has been added to enable the comparison of the combined aggregation / classification networks with standard classifiers.

A subset of three so-called *simple* data sets aims at measuring a candidate network's principal ability for information aggregation and has little immediate correspondence to practical applications. The five remaining data sets simulate classification tasks being typical for *surface inspection*.

3.2.1. Simple Data Sets

The structure of the feature data is identical for all three simple data sets. The data sets differ in terms of the rules used to generate the (good/bad) labels. The data sets contain 20.000 sets of real-valued 3-dimensional vectors. Each of the sets consists of a minimum of 3 and a maximum of 12 such vectors. The real-valued vector entries are randomly drawn from a uniform distribution in the interval [-0.5, 0.5]. Each vector is interpreted as a point in Euclidean 3-space.

Based on these point sets different kinds of aggregated features are computed for each data set. Image labels are then assigned by applying a threshold to the respective aggregated feature. The thresholds have been chosen to yield an even split of good/bad labels on each of the generated data sets.

- **Simple 1:** The aggregated feature used in the labeling rule is the mean value of the Euclidean norms in the set of feature vectors. Computing the mean value can be done incrementally and requires memory for the running mean value and counting the already processed entries. We consider this to be the base-line task for trainable feature aggregation.
- **Simple 2:** The aggregated feature is the sum of the two largest vector norms in the set. Like in the Simple 1 data set this task can be performed in a single step but it requires memory to store the two largest vector norms.
- **Simple 3:** The labeling rule in this data set is based on the minimum Euclidean distance between two points of the set corresponding to an image. Computing this rule is not possible in a single pass with constant memory. For classification of sets with arbitrary cardinality using finite memory the trainable aggregation function has to learn a heuristic leading to an approximate solution. However, since the cardinality of the vector sets in the test data is limited to 12, a finite network should be able to implement this kind of aggregation in an exact manner.

3.2.2. Surface Inspection Data Sets

Five data sets containing 20.000 images each have been generated. The images and the rules applied for labeling simulate typical tasks occurring in surface inspection applications. One bright spot within the image represents one object (potential fault) that is described by a feature vector. All of the objects in the image make up a set of faults that needs to be classified. Typical example images are displayed in Figure 4 and Figure 5. Both of them show preprocessed images from a surface inspection application. The background is removed and only relevant objects (potential faults) are presented in the image. The gray value corresponds to the degree of deviation from 'normal appearance' of the part of the image.

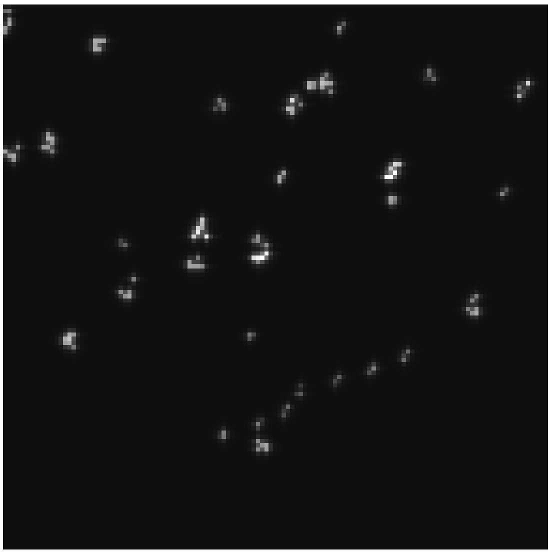


Figure 4: images from a surface inspection application with more evenly distributed flaws and line of flaws in the lower half.

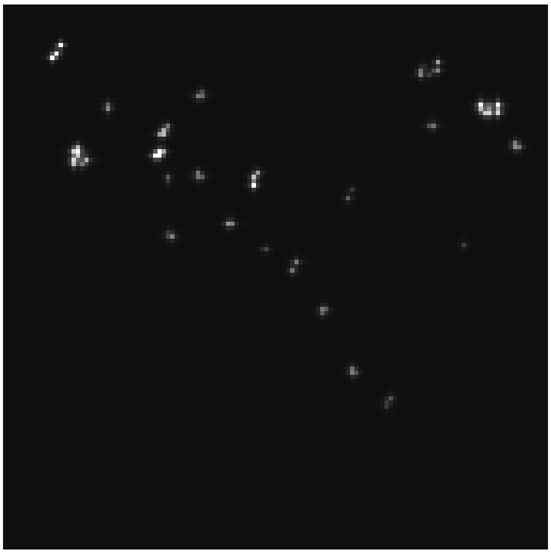


Figure 5: images from a surface inspection application with flaws concentrated in the upper half of the image except a line-like structure from the upper left to the lower left.

Both pictures show flaws of similar type and amount but a completely different distribution can be noticed. The flaws in Figure 1 are more evenly distributed, with some flaw clusters and some flaws apparently aligned. In contrast to that the flaws in

Figure 2 tend to remain in the upper half of the image and some of them build a nearly straight formation running from the upper left to the lower right.

The different composition of flaws leads to different properties of each image, respectively the corresponding feature sets.

The characteristic object features used to define the rules in the surface inspection data sets are:

- Center coordinates
- Area
- Compactness
- Maximum gray value
- Circumference
- Minimal and maximal extension

Based on these features rules to label images as good or bad are built by combining multiple sub rules with a logical OR operation. The sub rules correspond to simple threshold operations on object features and more importantly, on aggregated features. An example sub rule very common to surface inspection tasks combines object size (represented by area) with deviation from normal color (represented by maximum gray value):
An image is labeled bad if there is an object larger than 55 pixels AND maximum gray value is above 25.

All surface inspection data sets contain such sub rules based on object features only. Four surface inspection data sets additionally contain sub rules involving aggregate features, which should be learned by the aggregation part of our model. As already mentioned before, we have included one data set (Surface Inspection 1) not requiring aggregated information at all for correct classification. Below we describe the specific aggregated features used in sub rules of the data sets 2-5:

- **Surface Inspection 2:** The used rules to label the images contain two sub rules involving an aggregate feature, specifically the object count. *An image is labeled bad if there are more than 7 faults OR if there are more than 5 faults in the right half of the image.*
- **Surface Inspection 3:** The rules used to label the images contain two sub rules involving aggregate features based on counting and clustering. A cluster is defined as a number of objects concentrated in a circular area. *An image is labeled bad if there is a cluster of at least 4 objects within a 20 pixel radius OR if there are at least 3 clusters with 3 objects.*
- **Surface Inspection 4**: In this data set we have applied a sub rule based on object features to multiple objects, thus yielding a rule requiring aggregate features. *An image is labeled bad if there is an object larger than 10 pixels with maximum gray value above 150 AND there is another object larger than 15 pixels with maximum gray value above 100.*
- **Surface Inspection 5:** We consider this data set to contain the most difficult learnable aggregation rule in all given test data sets. By some means, the aggregation function has to test for compliance of a number of objects with a geometrical model: *An image is labeled bad if there are at least 6 objects aligned on a line, with the distance between object centers being smaller than 70 pixels and the orthogonal distance to the line being smaller than 15 pixels.*

4. EXPERIMENTS AND RESULTS

Due to the known principal limitations of standard classification methods like linear classifiers and feedforward neural networks in the field of classifying sets of objects with a priori not known number of objects and the known advantages of recurrent neural networks, we investigated different problems to compare recurrent neural networks to the standard classification methods. We therefore developed the settings Comp1 to Comp7 (Section 3).

4.1. Linear Classifier

The tests with linear classifiers where obtained as a benchmark for tests with feedforward neural networks as well as with recurrent neural networks. Due to the specific method, linear classifiers use, we arranged all feature vectors in one vector.

The percentage of falsely classified pictures is displayed in Table 1.

Table 1: Percentage of falsely classified pictures by linear classifiers

DS	crit. feat.	'sum of vector'	'areamax'	'areasum'	'number of flaws'	'minimal distance'	'cluster formation'	'number in region'
1	1	0	30.220	0	0	24.005	41.165	18.460
2	1	0	31.290	0	0	27.855	40.855	18.940
3	2	0	37.200	0	0.050	22.230	31.785	26.735
4	2	0	29.980	0.100	0.025	24.790	36.785	29.470
5	2	0	29.820	0	0.055	27.650	36.720	27.475
6	2	0	28.925	0	0.165	22.560	32.405	23.885
7	3	0.065	28.945	0.075	0.050	31.285	41.155	32.205
8	3	0.085	28.84	0	0.050	27.205	29.385	36.975
9	4	0	23.540	0.065	0.050	27.110	27.110	41.405
10	4	0.030	26.495	0	0.055	27.935	27.935	33.750

4.2. Feedforward neural networks

The architecture was chosen according to the test data. We used a network with 24 input neurons, one hidden layer with 6 neurons and one output neuron. As a training method a backpropagation method was used.

Results of classification are shown in Table 2.

Table 2: Percentage of falsely classified pictures by feedforward neural networks

DS	crit. / feat.	'sum of vector'	'areamax'	'areasum'	'number of flaws'	'minimal distance'	'cluster formation'	'number in region'
1	1	0.30	17.36	0.12	0.01	19.71	32.01	11.01
2	1	0.12	15.31	0.12	0.20	20.78	31.66	13.95
3	2	0.15	8.58	0.11	0.08	20.52	25.77	6.95
4	2	0.11	6.54	0.15	0.09	19.48	26.50	8.26
5	2	0.08	10.52	0.13	0.03	20.08	26.35	7.60
6	2	0.13	5.56	0.20	0.09	19.93	26.20	7.56
7	3	0.12	3.32	0.41	0.93	18.73	18.17	2.90
8	3	0.09	3.42	0.20	0.11	17.62	17.16	2.88
9	4	0.09	6.25	0.30	0.07	14.86	13.30	0.65
10	4	0.22	2.06	0.11	0.03	16.24	15.82	1.09

4.3. Recurrent neural networks

According to the structure of the networks, much smaller recurrent neural networks compared to the tests with feedforward neural networks have been used. Results are given in Table 3.

Table 3: Percentage of falsely classified pictures by recurrent neural networks

DS	crit. / feat.	'sum of vector'	'areamax'	'areasum'	'number of flaws'	'minimal distance'	'cluster formation'	'number in region'
1	1	8.37	35.05	29.49	27.77	22.74	40.30	13.88
2	1	23.86	33.53	43.37	37.41	20.84	36.90	27.81
3	2	25.82	40.89	26.96	6.95	33.67	29.27	34.92
4	2	20.06	44.63	18.93	10.88	21.91	28.89	22.84
5	2	14.99	45.51	17.61	44.11	23.34	27.90	16.52
6	2	19.96	36.27	27.28	4.20	20.95	30.92	10.16
7	3	16.68	45.18	28.12	6.76	21.21	29.99	19.26
8	3	15.61	39.94	25.29	14.72	33.78	23.57	14.76
9	4	12.95	44.38	24.32	8.54	29.62	31.88	9.56
10	4	13.98	43.61	15.29	10.03	29.71	20.99	10.10

Since some tasks where known as linearly separable, the good performance of the linear classifier was expected. For these problems nearly as good results were obtained with the feedforward neural network but it outperformed the linear classifier in the more complex problems partially dramatically.

The effort of the recurrent neural networks is twofold. On the one hand the results of classification are not that outstanding, especially for the linearly separable tasks. On the other hand, the effort in solving complex problems is comparable with results from the other methods, in particular if we consider the small number of neurons and therefore number of weights compared to the number of weights for feedforward neural networks.

Since these test scenarios did not have nearly the number of possible features of real surface recognition problems, the structural limitations of feedforward neural networks will probably eliminate them as a possible solution for such problems

5. OUTLOOK AND FUTURE WORK

Even though recurrent neural networks seem to be a good method to solve classification problems of complex type, significant and thorough research has to be done.

It is still unclear which training algorithm will give best results, as our backpropagation still has problems if the number of iterations is larger than 10. This poses the question of the best architecture in respect of the number of layers, neurons and iterations needed. The application of evolutionary computation methods like genetic algorithms and their combination with artificial neural networks gives a promising direction for future research.

To get more insight in the behavior and properties of recurrent neural networks, scenarios Simple 1 – Simple 3 and Surface Inspection 1 – Surface Inspection 5 will be investigated.

REFERENCES

Connor, J.T., Martin, D., Atlas, L.E. 1994. Recurrent neural networks and robust time series prediction, *IEEE Trans. on neural networks*. 5 (2): 240-254

Frasconi, P., Gori ,M., Sperduti, A. 1998. A general framework for adaptive processing of data structures. *IEEE Trans. On Neural Networks*, 9 (5)

Hammer, B., 2000. Hammer, B., 2001. Neural Networks Classifying Symbolic Data. *Proceedings of the ICML-2000 Workshop on Attribute-Value and Relational Learing: Crossing the Boundaries, 61-65*

Hammer, B., 2001. Generalization Ability of Folding Networks. *Knowledge and Data Engineering*, 13 (2): 196-206

Hammer, B., 2002. Recurrent networks for structured data – a unifying approach and its properties. *Cognitive Systems Research*, 3: 145-165

Hochreiter, S., Schmidhuber, J. 1997. Long short-term memory. *Neural Comp*. 9: 1735-1780

Ovchinnikov, S., Dukhovny, A. 2002. On order invariant aggregation functionals, *J. of Math. Psychology*, 46: 12-18

Perlich, C. Provost, F. 2003. Aggregation-based feature invention and relational concept classes, *Proc. 9th ACM SIGKDD international conference on knowledge discovery and data mining*. 167-176

Pollack, J.B. 1990. Recursive distributed representations, *Artificial Intelligence*, 46: 77-105

Ramon, J., De Raedt, L. 2000. Multi instance neural networks, *Proc. of the ICML workshop on attribute-value and relational learning*

Seichter S. 2006, Model-based Classifiers for Surface Inspection Problems, Diploma Thesis, *Vienna University of Technology*

Sperduti, A., Starita A., 1997. Supervised neural networks for the classification of structures. *IEEE Trans. On Neural Networks*, 8 (3): 714-735

Uwents, W., Blockeel, H. 2005. Classifying relational data with neural networks. *Proceedings (Kramer, S. and Pfahringer, B., eds.). Lecture Notes in Computer Science*. 384-396, Inductive Logic Programming, 15th International Conference, ILP 2005

Werbos, J. 1990. Backpropagation through time: What it does and how to do it, *Proc. IEEE.* 78 (10): 1550-1560

DISTRIBUTED DISCRETE SIMULATION ON THE WEB

Aman Atri[(a)], **Felix Breitenecker**[(b)], **Nicole Nagele**[(b)], **Shabnam Tauböck**[(c)]

[(a)(b)(c)(d)]Vienna University of Technology, Austria

[(a)]aman.atri@tuwien.ac.at, [(b)]felix.breitenecker@tuwien.ac.at, [(c)]nicole.nagele@tuwien.ac.at
[(d)]shabnam.tauboeck@tuwien.ac.at

ABSTRACT

With the vast development of internet technologies web based simulators have come to an extent where flexible loosely coupled components communicate in a distributed way. We are introducing a software architecture where the simulation environment is not limited to browser or applet specific restrictions but using the web as a transport layer. Furthermore this architecture permits us to separate the location of the data collection, the computational part and the visualisation modules. For a more transparent communication between distributed simulators a generic higher level semantics is able to pass information and data on top of a lower-level networking protocol. The idea is to design a system where modules can be deployed and interchanged without redesigning the whole simulator.

Keywords: web based simulation, service oriented simulation architecture, asynchronous communication, dynamic distributed code generation

1. INTRODUCTION

The rising facilities of internet based applications have brought a lot of new paradigms in terms of simulation software. The current trend of simulation frameworks and tools is to get rid of providing only locally restricted simulation environments and to seek for appropriate communication technology for interacting in heterogeneous systems.

Simulation tools can be used within given protocols like HTTP. The current development of web based technologies is leading the World Wide Web to a new era where internet is not only serving the purpose of displaying information in a browser but to interact and exchange information and data. The idea is not to re-invent new communication algorithms but to use existing transfer protocols and to build higher-level architectures which can provide a much more complex and powerful semantics. This paper deals with web based modelling strategies, communication between client and server applications and network solutions for building higher-level simulation environments for larger scalability (Atri 2007, Breitenecker 2006).

The rapid development of the World Wide Web has made it possible that data can now be transferred without constraints concerning target platforms or location of systems.

Distributed simulation on the web requires data exchange from remote components. Thus there will be a detailed view on certain software architectures that are suitable for these scenarios. Complications, problems that do occur in modelling and designing simulations scenarios are also introduced.

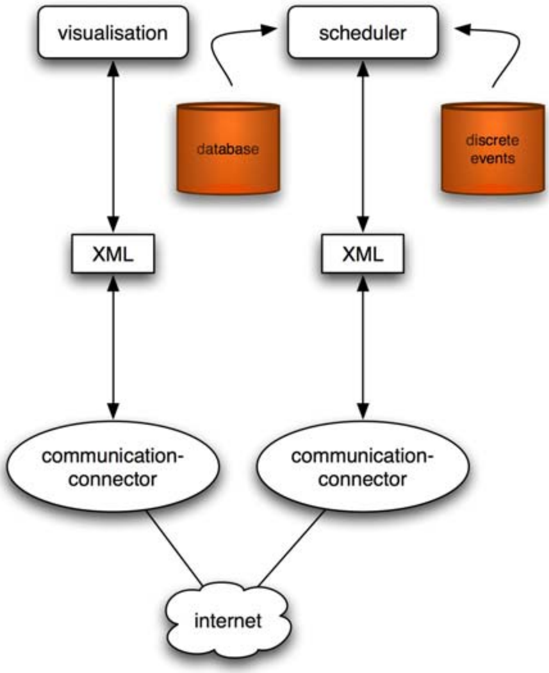


Figure 1: Message passing in transparent networks

2. HETEROGENEOUS COMMUNICATION

Distributed computation demands that our models are spread across the network and they share an internal communication channel to update themselves. Figure 1 shows a loosely coupled framework of a message passing layer which provides an abstraction regarding the dependency towards the actual programming and simulation language. Thus the actual target platform gets independent if we use a globally known and open communication standard like XML (Lechler 99). Now that most of the components do not necessarily share a common physical environment, unified arrangements must be made so that these components can broadcast their behaviour and presence to other subsystems. Depending on the actual modelling and experiment, web based simulation systems do not only work on broadcasting patterns but can also benefit from subscribing to certain services which are required by the simulation algorithm. This allows the whole system to provide the ability for remote computing within the given context.

2.1. Web based visualization

Due to the behaviour of classical stateless protocols like HTTP data transfer is restricted in direction, chronological order and as a negative side effect it produces a traffic overhead creating potential falsified discrete triggering data. This might result in latency effects in visualization on the client side. There are higher-level approaches of web programming frameworks to implement visualisation trough:

- Dynamical code generation with asynchronous data delivery from the server
- A statically pre-compiled graphical application which can directly access the service and might be able to overtake some portions of the computing unit

The first approach is implemented in Web 2.0 fashion using XML mapping between clients and server. Ajax (Asynchronous JavaScript and XML) emulates the characteristics of a stateless protocol. In case of required data update, the application is able to receive the new visualisation information in background without interrupting the actual simulation process.

The corresponding source code can be generated via the web application on the server side. On the other hand a predefined simulation and visualisation engine is preferred if the data representation is not manipulated from the distributed environment but can be computed on a local system. In the last years this practice has been put using Java Applets embedded in a browser but now it has become more comfortable to develop programs which use native graphical widgets of the operation system and still the application is platform independent. Java Webstart (JWS) simulation applications offer a fast way to synchronize and interact with the simulation service and download the models from the net dynamically (Page and Kreutzer 2005).

The interactive web based simulation enables the user to enter parameters or modify them using the browser. Usually if the application is not running only on the client side the browser sends this parameter to the web server using the HTTP protocol. The request is matched with the unique session of the user and the session values are updated with the latest parameters. The simulation web application is sends the result to the browser and the page is reloaded. Because of this latency during the HTTP requests the visualisation cannot run fluently. Ajax technology is a work around the avoid networking overhead. During the simulation process the data is not changed entirely. The static components which are displayed for documentation purposes remain always the same and only the meta-info graphical part is computed dynamically. For example if the simulation visualisation is represented as a graphical chart where the drawn function is changing its values in certain time intervals and the browser refreshes the page at every new query. All the information which never change like the websites layout style sheet, the cookies etc. will be resent over the network causing a falsified query time stamp because the browser will take its own time to reformat and redisplay the whole page.

If the components of the graphical chart are using Ajax JavaScript then the web application will only receive an XML formatted query where the variables which are necessary for repainting the graph are sent and recomputed without sending the whole HTTP request new. This communication is transported asynchronously and the user has not the responsibility to update and refresh his browser window. The document exists during the entire user session and is only modified within the permitted context. In combination with hyperlinks parts of the visualization can be hidden without reformatting and reading the style sheet commands. The value of the hyperlinks can also be changed dynamically and so they get a semantic quality. Dispatching Ajax queries does not require special browser plugins or extra ordinary configuration. Any modern web browser with enabled JavaScript is capable to deal with asynchronous XML communication. The web application which is computing the simulation results is of course aware of its Ajax capacity. In visualisation where the size of pending data is not known (the simulation time is known but the total amount of computed data might be unpredictable) Ajax helps the simulator to avoid unnecessary traffic overhead.

2.2. Architectures for discrete Simulation

Building simulation networks where unpredictable number of components work together or may drop out implies to design an architecture which implements the following criteria:

- The client side should be simple and contain only a few classes to decrease overhead traffic
- Discrete events triggered remotely have to be recognized and verified in case of data is lost or falsified during data transfer
- Fault tolerance mechanisms have to grant the exchangeability of components in case of loss of connectivity. The simulation engine has to be notified when some remote modules are not reachable

Event oriented systems interact not trough a stream of information and data but with synchronized events and require an architecture which is mostly suitable if we use the web as a global computation platform. The main focus is not only to optimize single algorithms but the whole system. Event oriented systems don’t interact trough a stream of information and data but with synchronized events and require an architecture which is mostly suitable if we use the web as a global computation platform for a frictionless integrity of all storage and computing units (Bass and Clements and , Kazman 1998).

Distributed simulation refers to a system where the components do not only reside on a single system. The communication is done over an internet protocol between the applications. A fault tolerant distributed application is a system where a single component might be out of order but not implicating a shutdown of the whole system. While the retrieval of the faulty component seems to be more or less easy, the actual difficulties rise when we have to look for an appropriate substitution.

2.3. Service-Oriented Simulation

When we talk of web based access to simulation resources we find a lot of end-to-end point implementations where the data is passed over an HTML site of a browser to a CGI script on a server and the computation is done either on the server and sent back to the browser of the client system or we get an inline plugin such as an applet or a flash animation. (Page and Lechler and Classen 2000) The limitation of a duplex client-server architecture is that most of the data might be hidden and we don't have the permission to access them directly. This is were services-oriented architecture comes into practice during a bidirectional message passing when both nodes have to behave not only as a server but also as a client. One implementation of a service-oriented architecture (SOA) is called web services which use an existing networking protocol like HTTP(s). The semantic layer is represented as a XML specification where important information like the names of the classes and methods which are likely to be executed remotely and the parameters and the return values are transferred. In discrete event oriented simulation a sequential concatenation is strategy to implement a distributed waiting queue area (Dustdar and Gall and Hauswirth 2003).

Thus such a simulator works on service-oriented pipes. That means, that every web service offers its functionality to another one. The output of a transaction can then be used as an input for another web service and so on. In case of failure it is very easy to diagnose which chain link is out of order and can be replaced by simply switching to another web service node and redirecting the stream.

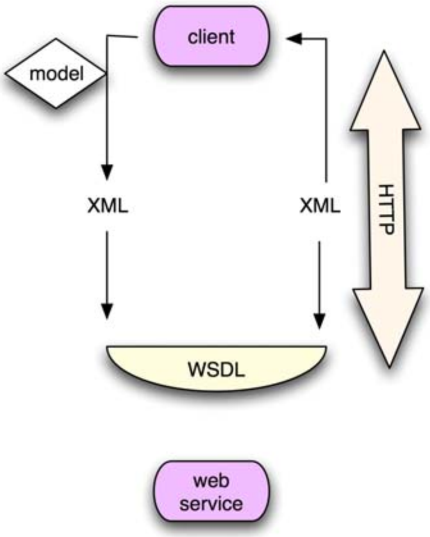


Figure 2: Transaction of a model over SOAP

One big advantage of this concept is the possibility that entities can be handled without depending on a specific programming language. The WSDL specification which provides information about the classes and the interfaces is adopted in most of object oriented languages and frameworks (Gyimesi 2005, Booth et. al. 2004). Especially in discrete simulation we can distinguish the state of a model by its embedded variables and their values. Web services can now provide a manipulation of these variables and programmatic computation without even knowing in which actual language they have been originally configured.

2.4. Sequential and Parallel Piping

Some processes during a discrete simulation computation may require merge and branching of the current model. In that case a web service oriented queue can split the model into several subsets of variables and pass these subsets to different sub-services.

A sequential pipe can branch out to a parallel pipe. That does not imply that the actual calculation is performed as a parallel process but more like a splitted workflow action where the order of the output is irrelevant. If such a service requires some variables from another set, but due to security restrictions the direct access is not granted, then the demanding web service can request a fetch task from the web service which is managing the referring data set. An end-point SOAP station which merges the model together and releases it to the next module is globally accessible.

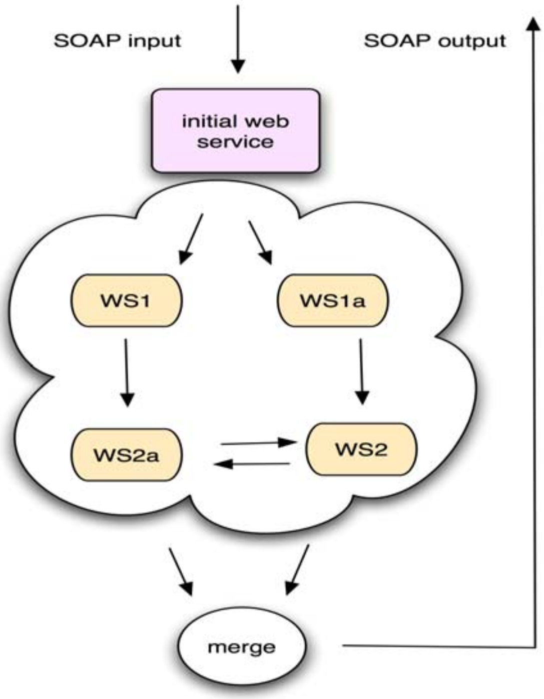


Figure 3: Task divided Simulation Web Services

For any module outside of this constellation the whole system seemed to be a black box. The waiting queue which entered the system had been split and passed on to several different host systems. The policy by which the data is distributed is depending on the current task of the discrete simulation. Such a policy could be driven by access permission of secondary data source which is essential for the computation or the positive side effect of efficient load balancing.

2.5. Multiple persistence of discrete models

In discrete event oriented simulation some tasks are likely to avoid temporal latencies because of quick read and write operations. In many cases the actual state of the model has to be available to all stakeholders in the network. This condition leads to a symbiosis where the following criteria have to be met:

- Every client gets an update of the state of all entities at any time requested.
- Any manipulation of the state of a model has to be transactional. That means every client has to be notified that an update has been committed. If a client does not respond positively than either the transaction has to be withdrawn or the corresponding client has to be listed as non-active.
- Every entity which is accessible to all clients has at least one copy of itself in the system.

2.6. Models in a tupel space

Discrete event oriented simulation works on the communication and message passing between the entities and their sinks and sources. In large scale networks this message passing could get in a bottleneck situation if the whole repository is centralized or the network traffic is unbalanced and some clients might experience latency effects. Furthermore if a centralized repository loses its network connection the whole simulation execution will get stuck. An appropriate mapped out persistence strategy is a so called tupel space as shown in Figure 4.

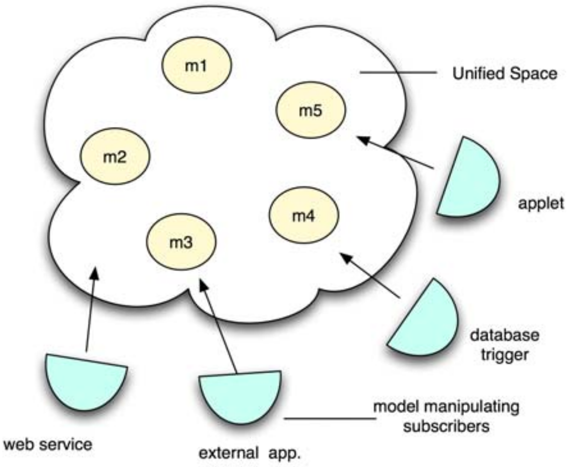


Figure 4: Multiple persistence in a unified space

Somehow in certain work steps the number of read and write operations cannot be predicted. The idea is to unify the memory of all stakeholders to a so called global unified tupel space (TS). Within this space the models are accessible to all memory-providing hosts and are replicated. This means that each model is handled as a unique object in comprehensive higher level memory architecture but actually the tupel space service provides the opportunity that in case of failure of certain parts of the network, the object and data will not be lost and other tupel service nodes will take over the data. Besides that a change of the state of a discrete model (e.g. a modification of the model parameters) will be passed on to every replicated copy in a synchronized fashion.

The programming paradigm of this higher lever architecture was introduced by Dr. David Gelernter at the Yale University as coordination language named Linda. Afterwards a Java based approach was implemented by Sun called Java Spaces. A Java Space service network does not offer a querying language as relational or object oriented database. In discrete simulation the selection of each model is conducted by rules which are based on discrete stochastic events or external temporal depending influences. The main focus of a tupel space is the correct identification of the objects for successful retrieval and transactional synchronisation. The lookup methods for models in space deliver two types of results: find the exact match or just forget it. A unified memory is quite comfortable for discrete event oriented models using cellular automata (CA) as their state change navigator from the programming point of view because modifications of the current state affecting the whole CA are automatically replicated and updated.

The disadvantage of using Java Spaces is that data transfer within the space itself and with the client is done using Java objects. But building a multi-tier architecture as shown in Figure 4 enables language independent access by inserting a middleware layer which reallocates interfaces like web services or hibernation mapping to external clients. Thus the tupel space can be accessed by any database client, web service client or even directly with an application written in Java like a web based applet or Java Webstart application.

2.7. Publishing and subscription vs. parametric selection

While tupel spaces favour the idea of distributed object storage without interfering with the semantic layer an alternative approach is to allow an external managing software module to allow publishing and subscribing (P/S) to certain objects which represent the state of the model.

Thus we can couple the tupel space storage with external business logic. The concept of P/S is comparable to a subscription system like a newsletter service. The broking system broadcasts its models according to its category. For instance if the discrete simulation is going to compute a distributed waiting queue, a P/S broking system could mange different queues within a total different simulation context. A client could subscribe to a certain queue claiming for notification of only specific types of entities loaded. This can be compared with the waiting queue at an airport. After the passenger enters the airport he will be redirected to the terminal to submit his luggage. This is would resemble our distributed queue. A subscription request is expressed by an airline only for their own passengers. Although the *middleware* (the waiting hall in front of the check-in counters) is managing all passengers of all airlines the counter of a categorized airline can then trigger a notification when a passenger travelling with the corresponding company delivers his luggage.

In object-oriented simulation over a network this pattern is very useful because the P/S broker can handle any arbitrary number of clients. Subscribers gain a profit because they don't have to take care of filtering the search results. They trust their own broker manager who is responsible for the correct selection of the models with the following properties:

- A global queue is storing the events
- Publishers register themselves at the global queue
- Publishers create new events if the state of the model changes
- Subscribers register themselves and describe those events they want to be informed about
- A broker manager who is hired by the subscriber observes the queue and the stored events
- In case the state of a model has been changed due to an event caused by the publisher the broker sends a message to all subscribed clients using either the push or pull method
- *Pull model*: the subscriber receives a notification that data has been changed. The subscription client is now responsible to fetch an update of the model
- *Push model*: the client does not only get the notification but also the whole data bundle.

In event oriented simulation with lots of models states and large objects the pull method reduces the network traffic overhead because the client can filter the content and update only parts of the model which are required for the next simulation process.

2.8. Persistence and Transformation of objects

Time consuming simulation with lot of input and output data require sophisticated persistence of the state of the models. The architectures discussed previously have introduced the exchange of information on different host systems but not the storage of those data models. In terms of large scale discrete simulations in an object oriented programming pattern the state of a discrete model is described by the values of its parameters methods and local and global variables. As a persistent storage platform an object oriented database would be the most appropriate solution. Many systems still do not support object oriented database persistence and store their information in relational databases because:

- Legacy software components would require the whole application to be rewritten
- Performance of relational databases are more efficient in terms of data mining and complex queries

Nevertheless dynamic models where the cardinality of their parameters can increase or decrease during simulation runtime, the object oriented database (OODB) can still recognize and verify these changes, as for the parameter of the model is stored as a complete serialized object. In a relational scheme this is not possible directly because of the static behaviour of the tables and their column definitions.

Hibernation technology plays a great middleware role when old database driven software meets new scalable pure object oriented design pattern. The layer of transforming an object into serialized data (which could be stored on a file system) is interrupted with a transformation of the objects encapsulated data into an

XML declaration of the variables and their visibility properties. Thus the object is transformed into a plaintext readable form where the semantic labels can be reinterpreted. This means that any database driven simulation library can store and read directly from the relational database while an object oriented client can transform (hibernate) the model into an XML form and convert it to SQL statements or into objects. Figure 5 shows the workflow of the hibernation process:

1. The simulator has been provided with a model description.
2. The simulator (or the single component if the simulation is distributed on a network) wants to gain access to data to feed the model.
3. The simulator forwards the model description to the hibernation middleware which has access to a relational database.
4. The hibernation server converts the request into SQL statements and executes them.
5. The result is converted into an object for the simulation client and sent back for further operations.

Thus the information that the data was actually stored in a relational scheme can be hidden and the simulator regards the object locally created instance.

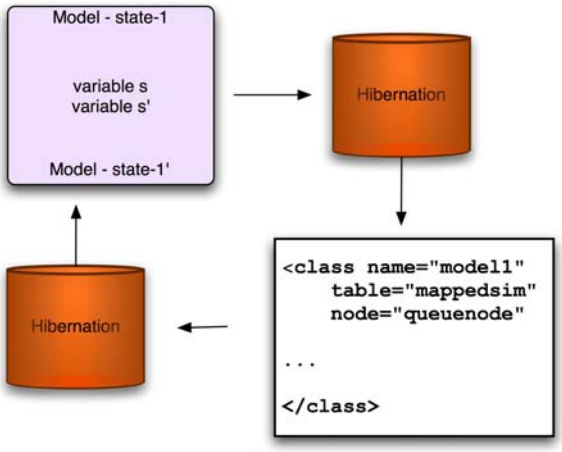


Figure 5: Dynamic mapping of model entities

3. CONCLUSION

We have shown that a composition of different architectural designs can build a highly scalable network for distributed simulation. Software architectures for discrete simulation are not merely focusing on parallel computation but also asynchronous exchange of the models and state modification. Higher level architectures can hide the actual machine representation of the model and operate only on the semantic value of the objects. Network transparent protocols can hide the complexity of the data transfer. For the simulator the actual location of the object is transparent and the models are treated as they would reside on the local system and the local runtime.

REFERENCES

Page, B., Kreutzer, W., 2005. *The Java Simulation Handbook. Simulating Discrete Event Systems with UML and Java*. Aachen:Shaker.

Page, B., Lechler T., Claassen S., 2000. *Objektorientierte Simulation in Java mit dem Framework DESMO-J Java.* Norderstedt:Books on Demand GmbH.

ARGE Simulation News, *ARGESIM Comparisons on Simulation Technique and Tools.* TU-Wien. Available from: http://www.argesim.org/comparisons [accessed 27 April 2008].

Atri, A., 2007. *Visualisierung verteilter diskreter Simulationen im Web*. Thesis (master). Vienna University of Technology.

Breitenecker, F., 2006. *Software for Modelling and Simulation - History, Developments Trends and Challenges.* Conference on Simulation and Visualization 2006, pp. 7-20. March 2-3 Magdeburg (Magdeburg, Germany).

Gyimesi, M., 2005. *Simulation Service Providing unter Verwendung von Web Service Technologie.* Thesis (PhD). Technische Universität Wien.

Booth D., Haas, H., McCabe, F., Newcomer E., Champion, I.M., Ferris, C., Orchard, D., 2004. *Web services architecture.* Technical report W3C – World Wide Web Consortium. Available from: http://www.w3.org/TR/ws-arch/ [accessed 27 April 2008]

Dustdar, S., Gall, H., Hauswirth M., 2003. *Software-Architekturen für Verteilte Systeme.* Berlin: Springer.

Lechler T., 1999. *Entwurf und Implementierung eines Frameworks für diskrete Simulatoren in Java*. Thesis (master), Universität Hamburg.

Bass L., Clements P., Kazman R., 1998. *Software Architecture in Practice*. Boston:Addison Wesley.

AUTHORS BIOGRAPHY

Aman Atri studied Software and Information Engineering at the Vienna University of Technology.

His bachelor thesis analyses automated proofs for model checking in temporal logic. After his bachelors program he pursued with the master course Software Engineering and Internet Computing. His master thesis deals with discrete simulation and visualisation schemes for the web. This work has been dilated in his PhD thesis where he is analysing and developing service oriented simulation frameworks and interoperable connectivity of different simulators. He is working as an assistant at the Vienna University of Technology (Institute for Analysis and Scientific Computing).

STRUCTURAL FEATURES IN SIMULATION SYSTEMS – EVOLUTION AND COMPARISON

Felix Breitenecker [(a)], **Nikolas Popper**[(b)], **Günther Zauner**[(a,b)]

[(a)] Vienna Univ. of Technology, Austria
[(b)] "die Drahtwarenhandlung" Simulation Services, Vienna, Austria

[(a)] Felix.Breitenecker@tuwien.ac.at, [(b)] Niki.Popper@drahtwarenhandlung.at

ABSTRACT

Object-oriented approaches, DAE modelling, variable structure modelling, Modelica notation and other developments have pushed the development of simulation languages essentially. A key point in these developments are hybrid structures and their related topics as events, event handling, state charts, structural changes, etc. This contribution compares features for modelling and simulation of extended hybrid structures in grown and new simulation systems on basis of classical features – model sorting, event description, event handling, DAE solver, index reduction, and on basis of structural features – physical modelling, Modelica modelling, state chart modelling, structural dynamic modelling, visualisation, environment. The comparison of these features is based on the ARGESIM Benchmarks for Modelling and Simulation Approaches, which investigate different model approaches and different simulation techniques by means of implementations in various simulators.

Keywords: Simulation software, CSSL standard, Modelica standard, hybrid structures, structural dynamic systems, feature comparison

1. ARGESIM BENCHMARKS ON MODELLING AND SIMULATION APPROACHES

In 1990, the journal ***SNE – Simulation News Europe*** – started a series on *Comparison of Simulation Software*, which has been developed to *Benchmarks for Modelling and Simulation Techniques*. Up to now, 20 comparisons and benchmarks have been defined, and about 250 solutions have been published – being a very valuable source for discussing and documenting various aspects of modelling and simulation approaches.

For the evaluation and comparing features and approaches solutions to these comparisons were used, mainly of the following comparisons:

- C 1 - Lithium-Cluster Dynamics under Electron Bombardment
- C 3 - Analysis of a Generalized Class-E Amplifier
- C 5 - Two State Model
- C 7 - Constrained Pendulum
- C 9 - Fuzzy Control of a Two Tank System
- CP1 - Parallel Comparison
- C 11 - SCARA Robot
- C 12 - Collision Processes in Rows of Spheres
- C 13 - Crane Crab with Embedded Control
- C 15 - Clearance Identification
- C 17 - Spatial Dynamics of SIR-Type Epidemic
- C 18 - Neural Networks versus Transfer Functions - Identification of Nonlinear Systems
- C 19 - Pollution in Groundwater Flow

.

This contribution mainly concentrates on Benchmark C5 *Constrained Pendulum*, because is is a small model and comparison results can be documented in a concentrated manner. At present, further benchmarks are in preparation, among them an extended benchmark for hybrid systems. Detailed information about definitions and solutions to these benchmarks can be found in **SNE**, www.argesim.org.

2. CLASSICAL FEATURES OF SIMULATORS

2.1 CSSL Structure for Simulators

In 1968, the CSSL standard set first challenges for features of simulation systems, defining necessary basic features for simulators and a certain structure for simulators (Figure 1).

The CSSL standard suggests structures and features for a model frame and for an experimental frame. This distinction is based on Zeigler's concept of a strict separation of these two frames. Model frame and experimental frame are the user interfaces for the heart of the simulation system, for the simulator kernel or simulation engine. A translator maps the model description of the model frame into state space notation, which is used by the simulation engine solving the system governing ODEs. This basic structure of a simulator is illustrated in Figure 1; an extended structure with service of discrete elements is given in Figure 2.

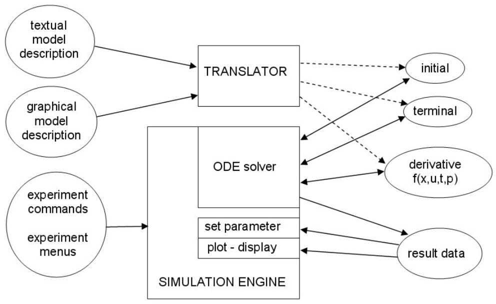


Figure 1: Basic Structure of a Simulator - CSSL

Between 1980 and 2000 developers put main emphasis on integration of discrete elements into continuous simulation systems, from simple time events to complex state events, and on extending the model description to DAEs. Both extensions are related, because algebraic equations are mainly caused or causing state events by means of state constraints.

Consequently, event description (ED), time event handling (TEH), state event handling (SEH) and DAE support by means of direct or iterative DAE solvers (DAE) with or without index reduction (IR) became desirable *Classical Features* of simulators, supported directly or indirectly – features to be discussed in more details in the next subsections.

2.2 Implicit Models – Differential-Algebraic Equations – DAE Solvers

For a long time the explicit state space description

$$\dot{\vec{x}}(t) = \vec{f}(\vec{x}(t),\vec{u}(t),t,\vec{p}), \quad \vec{x}(t_0) = \vec{x}_0 \tag{1}$$

played the dominant role; additional constraints and implicit models had to be transformed 'manually'. From the 1990s on, the simulators started to take care on these very natural phenomena of implicit structures. Consequently, they started to deal with implicit state space descriptions and constraints, in general with so-called DAE models (differential algebraic equations):

$$F(\dot{\vec{y}}(t),\vec{y}(t),\vec{u}(t),t,\vec{p}) = \vec{0} \quad \vec{y}(t_0) = \vec{y}_0 \tag{2}$$

The so-called extended state vector $\vec{y}(t)$ can be splitted into the differential state vectors $\vec{x}(t)$ and into the algebraic state vector $\vec{z}(t)$:

$$\begin{aligned} &\dot{\vec{x}}(t) = \vec{f}(\vec{x}(t),\vec{z}(t),\vec{u}(t),t,\vec{p}) = 0, \quad x(t_0) = \vec{x}_0, \\ &\quad g(\vec{x}(t),\vec{z}(t),\vec{u}(t),t,\vec{p}) = 0 \end{aligned} \tag{3}$$

The above given DAEs can be solved by extended ODE solvers and by implicit DAE solvers. Three different approaches may be used:

1. *Nested Approach*, using classical ODE solver
   (a) given x_n, solving first numerically $g(x_n,z_n)=0 \Rightarrow z_n = z_n(x_n) = \hat{g}^{-1}(x_n)$, e. g. by modified Newton iteration, and
   (b) applying ODE method, evolving $x_{n+1} = \Phi_E(x_n, z_n(x_n), t_n)$.
2. *Simultaneous Approach*, using an implicit DAE solver; given x_n, solving $g(x_{n+1},z_{n+1})=0$, $\Phi_I(x_{n+1},x_n,z_{n+1},t_{n+1})=0$ simultaneously.
3. *Symbolic Approach*, determining in advance the explicit form solving $g(x,z)=0 \Rightarrow z = z(x) = \hat{g}^{-1}(x)$ by symbolic computations e.g. within the model translator, and using classical ODE solvers.

The *Symbolic Approach* requires a symbolic inversion of the algebraic equations, which in many cases is not possible or not adequate; furthermore the model translator must not only sort equations, it must be able to perform symbolic manipulations on the equations.

The *Nested Approach* – up to now most commonly used – requires a numerical inversion of the algebraic equations: each evaluation of the vector of derivatives (called by the ODE solver) has to start an iterative procedure to solve the algebraic equation. This approach can be very expensive and time-consuming due to these inner iterations. Here classical ODE solvers can be used.

The *Simultaneous Approach* requires an implicit ODE solver – usually an implicit stiff equation solver. Although also working with iterations, these solvers show much more efficiency and provide more flexibility for modelling (DASSL, IDA-DASSL, and LSODE – solvers).

However, hidden is another problem: the 'DAE index' problem. Roughly speaking, a DAE model is of index *n*, if n differentiations of the DAE result in an ODE system (with an increased state space). The implicit ODE solvers for the *Simultaneous Approach* guarantee convergence only in case of DAE index $n = 1$. Models with higher DAE index must / should be transformed to models with DAE index $n = 1$. This transformation is based on symbolic differentiation and symbolic manipulation of the high index DAE system, and there is no unique solution to this index reduction.

The perhaps most efficient procedure is the so-called *Pantelides Algorithm.* Unfortunately, in case of mechanical systems modelling and in case of process technology modelling indeed DAE models with DAE index $n = 3$ may occur, so that index reduction may be necessary. Index reduction is a new challenge for the translator of simulators, and still point of discussion.

In graphical model descriptions, implicit model structures are known since long time as algebraic loops: the directed graph of signals has one or more signal feedback loops without any memory operator (integrator, delay, etc). Again, in evaluating the problem of sorting occurs, and the model translator cannot build up the sequence for calculating the derivative vector.

Some simulators, e.g. SIMULINK, recognise algebraic loops and treat them as implicit models. When a graphical model contains an algebraic loop, SIMULINK calls a loop solving routine at each time step - SIMULINK makes use of the *Nested Approach* described before. This procedure works well in case of models with DAE index $n = 1$, for higher index problems may occur.

In object-oriented simulation systems, like in Dymola, physical a-causal modelling plays an important role, which results in DAEs with sometimes higher index. These systems put emphasis on index reduction (in the translator) to DAEs with index $n = 1$ in order to apply implicit ODE solvers (*Simultaneous Approach*

2.3 Time Events and State Events

The CSSL standard also defines segments for discrete actions, first mainly used for modelling discrete control. So-called DISCRETE regions or sections manage the communication between discrete and continuous world and compute the discrete model parts.

For incorporating discrete actions, the simulation engine must interrupt the ODE solver and handle the event. For generality, efficient implementations set up and handle event lists, representing the time instants of discrete actions and the calculations associated with the action, where in-between consecutive discrete actions the ODE solver is to be called.

In order to incorporate DAEs and discrete elements, the simulator's translator must now extract from the model description the dynamic differential equations (derivative), the dynamic algebraic equations (algebraic), and the events (event i) with static algebraic equations and event time, as given in Figure 2 (extended structure of a simulation language due to CSSL standard). In principle, initial equations, parameter equations and terminal equations (initial, terminal) are special cases of events at time $t = 0$ and terminal time. Some simulators make use of a modified structure, which puts all discrete actions into one event module, where CASE - constructs distinguish between the different events.
These so-called *time events* are known in advance, so that scheduling of the time events can be handled easily, e.g. in the same manner than simulators schedule output events.

Much more complicated, but defined in CSSL, are the so-called *state events*. Here, a discrete action takes place at a time instant, which is not known in advance, it is only known as a function of the states.

For state events, the classical state space description is extended by the so-called state event function $h(\vec{x}(t),\vec{u}(t),\vec{p})$, the zero of which determines the event:

$$\begin{aligned}&\vec{x}(t) = \vec{f}(\vec{x}(t),\vec{u}(t),\vec{p},t),\\ &h(\vec{x}(t),\vec{u}(t),\vec{p},t) = 0\end{aligned} \tag{4}$$

Generally, state events (SE) can be classified in four types:

Type 1 – parameters change discontinuously (**SE-P**),
Type 2 - inputs change discontinuously (**SE-I**),
Type 3 - states change discontinuously (**SE-S**), and
Type 4 - state vector dimension changes (**SE-D**), including total change of model equations.

State events type 1 (**SE-P**) could also be formulated by means of IF-THEN-ELSE constructs and by switches in graphical model descriptions, without synchronisation with the ODE solver. The necessity of a state event formulation depends on the accuracy wanted. Big changes in parameters may cause problems for ODE solvers with stepsize control.

State events of type 3 (**SE-S**) are essential state events. They must be located, transformed into a time event, and modelled in discrete model parts.

State events of type 4 (**SE-D**) are also essential ones. In principle, they are associated with hybrid modelling: models following each other in consecutive order build up a sequence of dynamic processes. And consequently, the structure of the model itself is dynamic; these so-called structural dynamic systems are at present (2008) discussion of extensions to Modelica, see next chapters.

State events of type 2 (**SE-I**) are not really state events, they are time events. They are usually put in the list of state events, if a synchronisation of the ODE solver with an input jump should be forced.

As example, we consider the pendulum with constraints (*Constrained Pendulum*). Let φ define pendulum angle, and l, m and d parameters for length, mass, and damping. If the pendulum is swinging, it may hit a pin positioned at angle φ_p with distance l_p from the point of suspension. In this case, the pendulum swings on with the position of the pin as new point of rotation. The shortened length is $l_s = l - l_p$. and the angular velocity $\dot{\varphi}$ is changed from $\dot{\varphi}$ to $\dot{\varphi}\cdot l/l_s$ at position φ_p, etc. These discontinuous changes are state events, not known in advance.

With event function notation, the model for *Constrained Pendulum* is given by

$$\begin{aligned}&\dot{\varphi}_1 = \varphi_2,\ \ \dot{\varphi}_2 = -\frac{g}{l}\sin\varphi_1 - \frac{d}{m}\varphi_2,\\ &h(\varphi_1,\varphi_2) = \varphi_1 - \varphi_p = 0\end{aligned} \tag{5}$$

The example involves two different events: change of length parameter (**SE-P**), and change of state (**SE-S**), i.e. angular velocity).
The handling of a state event requires four steps:

1. Detection of the event, usually by checking the change of the sign of *h(x)* within the solver step over $[t_i, t_{i+1}]$
2. Localisation of the event by a proper algorithm determining the time t^* when the event occurs and performing the last solver step over $[t_i, t^*]$
3. Service of the event: calculating / setting new parameters, inputs and states; switching to new equations
4. Restart of the ODE solver at time t^* with solver step over $[\, t^*= t_{i+1}, t_{i+2}]$

State events are facing simulators with severe problems. Up to now, the simulation engine had to call independent algorithms, now a root finder for the state event function *h* needs results from the ODE solver, and the ODE solver calls the root finder by checking the sign of *h*. For finding the root of the state event function *h(x)*, either interpolative algorithms (MATLAB/Simulink) or iterative algorithms are used (ACSL, Dymola).

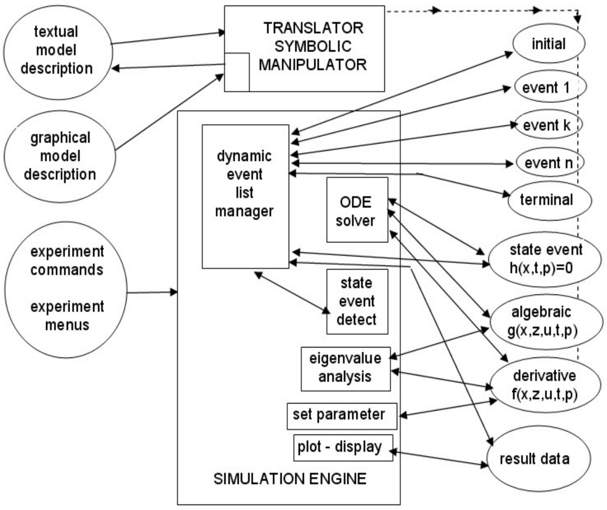


Figure 2: Extended Structure of a Simulation System due to Extensions of the CSSL Standard with Discrete Elements and with DAE Models.

Figure 2 (extended structure of a simulation language due to CSSL standard) also shows the necessary extensions for incorporating state events. The simulator's translator must extract from the model description additionally the state event functions (state event j) with the associated event action – only one state event shown in the figure). In the simulator kernel, the static event management must be made dynamically: state events are dynamically handled and transformed to time events. In principle, the kernel of the simulation engine has become an event handler, managing a complex event list with feedbacks. It is to be noted, that different state events may influence each other, if they are close in time – in worst case, the event finders run in a deadlock. Again, modified implementations are found. It makes sense to separate the module for state event function and the module for the associated event – which may be a single module, or which may be put into a common time event module.

In case of a structural change of the system equations (state event of type 4 – **SE-D**), simulators usually can manage only fixed structures of the state space. The technique used is to 'freeze' the states that are bound by conditions causing the event. In case of a complete change of equations, both systems are calculated together, freezing one according to the event. One way around is to make use of the experimental frame: the simulation engine only detects and localises the event, and updates the system until the event time. Then control is given back to the experimental frame. The state event is now serviced in the experimental frame, using features of the environment. Then a new simulation run is restarted (modelling of the structural changes in the experimental frame).

The *Constrained Pendulum* example involves a state event of type 1 (**SE-P**) and type 3 (**SE-S**). A classical ACSL model description works with two discrete sections `hit` and `leave`, representing the two different modes, both called from the dynamic equations in the derivative section (Table 1).

Dymola defines events and their scheduling implicitly by WHEN – or IF - constructs in the dynamic model description, in case of the discussed example e.g. by

```
 WHEN phi-phip=0
      AND phi>phip
THEN l = ls;
     dphi = dphi*lf/ls
```

In case of more complex event descriptions, the WHEN – or IF – clauses are put into an ALGORITHM section similar to ACSL's DISCRETE section.

Table 1: *Constrained Pendulum*: Continuous Model with State Events (ACSL)

```
PROGRAM constrained pendulum
CONSTANT m = 1.02, g = 9.81, d =0.2
CONSTANT lf=1, lp=0.7
DERIVATIVE dynamics
  ddphi = -g*sin(phi)/l – d*dphi/m
  dphi  = integ ( ddphi, dphi0)
  phi   = integ ( dphi, phi0)
  SCHEDULE hit   .XN. (phi-phip)
  SCHEDULE leave .XP. (phi-phip)
END ! of dynamics

DISCRETE hit
  l = ls; dphi = dphi*lf/ls
END ! of hit

DISCRETE leave
  l = lf; dphi = dphi*ls/lf
END ! of leave

END ! of constrained pendulum
```

In graphical model descriptions, we are faced with the problem that calculations at discrete time instants are difficult to formulate. For the detection of the event, SIMULINK provides the `HIT CROSSING` block (in new Simulink version implicitly defined). This block starts state event detection (interpolation method) depending on the input, the state event function, and outputs a trigger signal, which may call a triggered subsystem servicing the event.
It is to be noted, that discrete elements with time events and state events and DAEs may also change the structure of the model.

2.4 Classical Features of Simulators

Event description (ED), time event handling (THE), state event handling (SEH) and DAE support (DAE) with or without index reduction (IR) became desirable structural features of simulators, supported directly or indirectly. Table 2 compares the availability of these features in the MATLAB / Simulink System, in ACSL and in Dimple, based mainly on evaluations of the ARGESIM Benchmarks.

Table 2: Comparison of Simulators' Classical Features

	MS - Model Sorting	ED -Event Description	TEH – Time Event handling	SHE - State Event Handling	DAE - DAE Solver	IR - Index Reduction
MATLAB	no	no	no	(yes)	(yes)	no
Simulink	yes	(yes)	yes	(yes)	(yes)	no
MATLAB / Simulink	yes	yes	yes	yes	(yes)	no
ACSL	yes	yes	yes	yes	yes	no
Dymola	yes	yes	yes	yes	yes	yes

In Table 2, the availability of features is indicated by 'yes' and 'no'; a 'yes' in parenthesis '(yes)' means, that the feature is complex to use. MS - 'Model Sorting', is a standard feature of a simulator – but missing in MATLAB (in principle, MATLAB cannot be called a simulator). On the other hand, MATLAB's ODE solvers offer limited features for DAEs (systems with mass matrix) and an integration stop on event condition, so that SHE and DAE get a ('yes'). In Simulink, event descriptions are possible by means of triggered subsystems, so that ED gets a '(yes)' because of complexity. A combination of MATLAB and Simulink suggest putting the event description and handling at MATLAB level, so that ED and SHE get both a 'yes'. DAE solving is based on modified ODE solvers, using the nested approach (see before), so DE gets only a '(yes)' for all MATLAB/Simulink combinations. Time events are not supported in MATLAB, but they are basic feature in Simulink.

ACSL is a classical simulator with sophisticated state event handling, and since version 10 (2001) DAEs can be modelled directly by the residuum construct, and they are solved by the DASSL algorithm (a well-known direct DAE solver, based on the simultaneous approach), or by modified ODE solvers (nested approach) – so 'yes' for ED, SHE, and DAE. In case of DAE index $n = 1$, the DASSL algorithm guarantees convergence, in case of higher index integration may fail. ACSL does not perform index reduction (IR 'no'). ACSL comes with a sophisticated state event handling, so that all kind of events can be modelled and handled in a comfortable manner.

Dymola is a modern simulator, implemented in C, and based on physical modelling. Model description may be given by implicit laws, symbolic manipulations extract a proper ODE or DAE state space system, with index reduction for high index DAE systems – all classical features are available. Dymola started a new area in modelling and simulation of continuous and hybrid systems (see Section 3).

3. STRUCTURAL FEATURES IN SIMULATORS

There are three basic developments to extend the structure of simulators. First, the extension from ODEs to DAE stimulated the evolvement of *Physical Modelling* – modelling based on laws and physical 'modules', textually und graphically – Dymola started the development. Second, influences from computer engineering suggest use of UML – *Unified Modelling Language*, especially UML the use of UML state charts for discrete events. And third, as consequence of the hybrid decomposition of models by state charts, and influenced by experiences from co-simulation, handling of Structural Dynamic Systems became important.

3.1 Physical Modelling

In the 1990s, many attempts have been made to improve and to extend the CSSL structure, especially for the task of mathematical modelling. The basic problem was the state space description, which limited the construction of modular and flexible modelling libraries. Two developments helped to overcome this problem. On modelling level, the idea of physical modelling gave new input, and on implementation level, the object-oriented view helped to leave the constraints of input/output relations.

In physical modelling, a typical procedure for modelling is to cut a system into subsystems and to account for the behaviour at the interfaces. Balances of mass, energy and momentum and material equations model each subsystem. The complete model is obtained by combining the descriptions of the subsystems and the interfaces. This approach requires a modelling paradigm different to classical input/output modelling. A model is considered as a constraint between system variables, which leads naturally to DAE descriptions. The approach is very convenient for building reusable model libraries.

In 1996, the situation was thus similar to the mid 1960s when CSSL was defined as a unification of the techniques and ideas of many different simulation programs. An international effort was initiated in September 1996 for bringing together expertise in object-oriented physical modelling (port based modelling) and defining a modern uniform modelling language – mainly driven by the developers of Dymola. The new modelling language is called *Modelica*. Modelica is intended for modelling within many application domains such as electrical circuits, multibody systems, drive trains, hydraulics, thermo-dynamical systems, and chemical processes etc. It supports several modelling formalisms: ordinary differential equations, differential-algebraic equations, bond graphs, finite state automata, and Petri nets etc.

Modelica is intended to serve as a standard format so that models arising in different domains can be exchanged between tools and users. Modelica is a not a simulator, Modelica is a modelling language, supporting and generating mathematical models in physical domains.

When the development of Modelica started, also a competitive development, the extension of VHDL towards VHDL-AMS was initiated. Both modelling languages aimed for general-purpose use, but VHDL-AMS mainly addresses circuit design, and Modelica covers the broader area of physical modelling; modelling constructs such as Petri nets and finite automata could broaden the application area, as soon as suitable simulators can read the model definitions.

Modelica offers a textual and graphical modelling concept, where the connections of physical blocks are bidirectional physical couplings, and not directed flow. An example demonstrates how drive trains are modelled. The drive train consists of four inertias and three clutches, where the clutches are controlled by input signals (Figure 3). The graphical model layout corresponds with a textual model representation, shown in Table 3 (abbreviated, simplified).

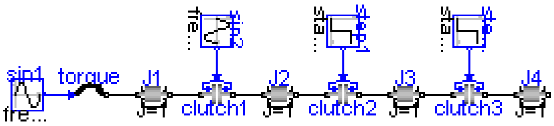


Figure 3: Graphical Modelica Model for Coupled Clutches

Table 3: Textual Modelica Model for Coupled Clutches

```
encapsulated model CoupledClutches; "Drive train"
  parameter SI.Frequency freqHz=0.2; ….
  Rotational.Inertia J1(J=1,phi(ic=0),w(ic=10));
  Rotational.Torque torque;
  Rotational.Clutch clutch1(peak=1.1, fn_max=20);
  Rotational.Inertia J3(J=1); ……………………………………
equation
  connect(sin1.outPort, torque.inPort);
  connect(torque.flange_b, J1.flange_a);
  connect(J1.flange_b, clutch1.flange_a);
    ……………………………………..
  connect(step2.outPort, clutch3.inPort);
end CoupledClutches;
```

The translator from Modelica into the target simulator must not only be able to sort equations, it must be able to process the implicit equations symbolically and to perform DAE index reduction (or a way around).

Up to now – similar to VHDL-AMS – some simulation systems understand Modelica (2008; generic – new simulator with Modelica modelling, extension - Modelica modelling interface for existing simulator):

- Dymola from Dynasim (generic),
- MathModelica from MathCore Engineering (generic)
- SimulationX from ISI (generic/extension)
- Scilab/Scicos (extension)
- MapleSim (extension, announced)
- Open Modelica - since 2004 the University of Lyngby develops an provides an open Modelica simulation environment (generic),
- Mosilab - Fraunhofer Gesellschaft develops a generic Modelica simulator, which supports dynamic variable structures (generic)
- Dymola / Modelica blocks in Simulink

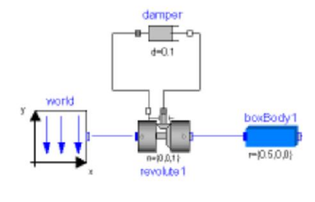
As Modelica also incorporates graphical model elements, the user may choose between textual modelling, graphical modelling, and modelling using elements from an application library. Furthermore, graphical and textual modelling may be mixed in various kinds. The minimal modelling environment is a text editor; a comfortable modelling environment offers a graphical modelling editor.

The *Constrained Pendulum* example can be formulated in Modelica textually as a physical law for angular acceleration. The event with parameter change is put into an `algorithm` section, defining and scheduling the parameter event **SE-P** (Table 4). As instead of angular velocity, the tangential velocity is used as state variable, the second state event **SE-S** 'vanishes'.

Table 4: Textual Modelica Model for *Constrained Pendulum*

```
equation /*pendulum*/
  v = length*der(phi);
  vdot = der(v);
  mass*vdot/length + mass*g*sin(phi)
  +damping*v = 0;
algorithm
 if (phi<=phipin) then length:=ls; end if;
 if (phi>phipin) then length:=l1; end if;
```

Modelica allows combining textual and graphical modelling. For the Constrained *Pendulum* example, the basic physical dynamics could be modelled graphically with joint and mass elements, and the event of length change is described in an `algorithm` section, with variables interfacing to the predefined variables in the graphical model part (Figure 4).



```
algorithm
if (revolute1.phi
     <= phipin then
    revolute1.length:=ls;
end if;
if (revolute1.phi
     < phipin then
    revolute1.length:=ll;
end if;
```

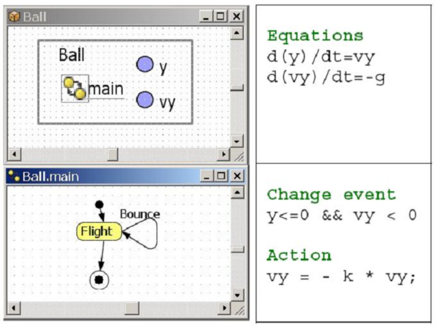
Figure 4: Mixed Graphical and Textual Dymola Model for *Constrained Pendulum*

3.2 UML State Chart Modelling

In the end of the 1990s, computer science initiated a new development for modelling discontinuous changes. The *Unified Modelling Language* (UML) is one of the most important standards for specification and design of object oriented systems. This standard was tuned for real time applications in the form of a new proposal, *UML Real-Time* (UML-RT). By means of UML-RT, objects can hold the dynamic behaviour of an ODE.

In 1999, a simulation research group at the Technical University of St. Petersburg used this approach in combination with a hybrid state machine for the development of a hybrid simulator (*MVS*), from 2000 on available commercially as simulator *AnyLogic*. The modelling language of AnyLogic is an extension of UML-RT; the main building block is the *Active Object*. Active objects have internal structure and behaviour, and allow encapsulating of other objects to any desired depth. Active objects interact with their surroundings through boundary objects: ports for discrete communication, and variables for continuous communication. The activities within an object are usually defined by state charts (extended state machine). While discrete model parts are described state charts, events, timers and messages, the continuous models are described by ODEs and DAEs in CSSL-type notation and with state charts within an object.

An AnyLogic implementation of the well-known *Bouncing Ball* example shows a simple use of state chart modelling (Figure 5). The model equations are defined in the active object ball, together with the state chart ball.main. This state chart describes the interruption of the state flight (without any equations) by the event bounce (**SE-P** and **SE-S** event) defined by condition and action.

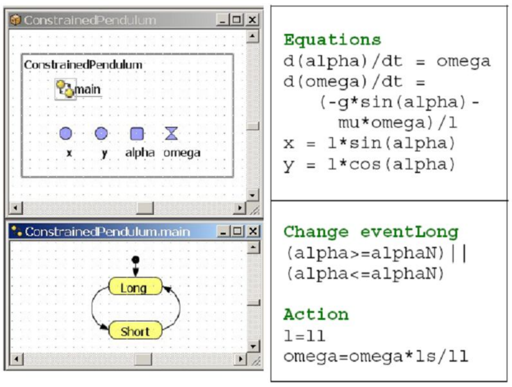


```
Equations
d(y)/dt=vy
d(vy)/dt=-g

Change event
y<=0 && vy < 0

Action
vy = - k * vy;
```

Figure 5: AnyLogic Model for the *Bouncing Ball*

AnyLogic influenced further developments for hybrid and structural dynamic systems, and led to a discussion in the Modelica community with respect to a proper implementation of state charts in Modelica. State charts are to be seen as comfortable way to describe complex WHEN – and IF – constructs, being part of the model, but state charts may also control different models from a higher level. A minor problem is the fact, that the state chart notation is not really standardised; AnyLogic makes use of the Harel state chart type.

An AnyLogic implementation for the *Constrained Pendulum* may follow the implementation for the bouncing ball (Figure 5). An primary active object (Constrained Pendulum)'holds' the equations for the pendulum, together with a state chart (main) switching between short and long pendulum. The state chart nodes are empty; the arcs define the events (Figure 6). Internally, AnyLogic restarts at each hit the same pendulum model (trivial hybrid decomposition).



```
Equations
d(alpha)/dt = omega
d(omega)/dt =
    (-g*sin(alpha)-
      mu*omega)/l
x = l*sin(alpha)
y = l*cos(alpha)

Change eventLong
(alpha>=alphaN)||
(alpha<=alphaN)

Action
l=l1
omega=omega*ls/l1
```

Figure 6: AnyLogic model for *Constrained Pendulum*, Simple Implementation

3.3 Structural Dynamic Systems

Hybrid systems – systems with state events of essential types, often come together with a change of the dimension of the state space, then called *Structural-dynamic Systems*. The dynamic change of the state space is caused by a state event of type **SE-D**. In contrary to state events **SE-P** and **SE-S**, states and derivatives may change continuously and differentiable in case of structural change. In principle, structural-dynamic systems can be seen from two extreme viewpoints. The one says, in a maximal state space, state events switch on and off algebraic conditions, which freeze certain states for certain periods. The other one says that a global discrete state space controls local models with fixed state spaces, whereby the local models may be also discrete or static.

These viewpoints derive two different approaches for structural dynamic systems modelling, the

- maximal state space, and the
- hybrid decomposition.

Most implementations of physically based model descriptions support a big monolithic model description, derived from laws, ODEs, DAEs, state event functions and *internal events*. The state space is maximal and static, index reduction in combination with constraints keep a consistent state space. For instance, Dymola, OpenModelica, and VHDL-AMS follow this approach.

The hybrid decomposition approach makes use of state events, which control the sequence and the serial coupling of one model or of more models. A convenient tool for switching between models is a state chart, driven by these events, which itself are generated by the models. Following e.g. the UML-RT notation, control for continuous models and for discrete actions can by modelled by state charts. This approach additionally allows not only dynamically changing state spaces, but also different model types, like ODEs, linear ODEs (to be analysed by linear theory), PDEs, co-simulation, etc. to be processed in serial or also in parallel, so that also co-simulation can be formulated based on external events.

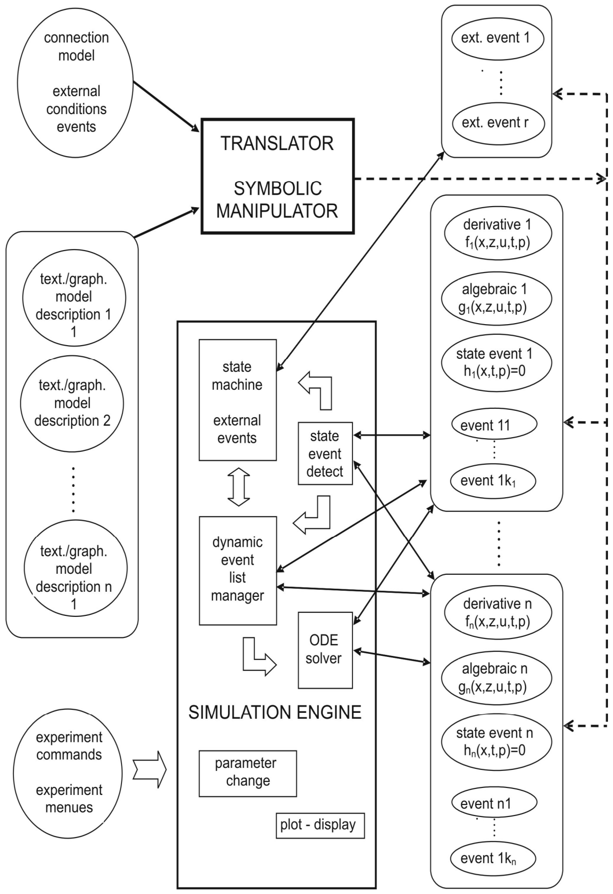

connection model
external conditions events
TRANSLATOR
SYMBOLIC MANIPULATOR
text./graph. model description 1 1
text./graph. model description 2
text./graph. model description n 1
state machine
external events
state event detect
dynamic event list manager
ODE solver
SIMULATION ENGINE
parameter change
plot - display
experiment commands
experiment menues
ext. event 1
ext. event r
derivative 1 $f_1(x,z,u,t,p)$
algebraic 1 $g_1(x,z,u,t,p)$
state event 1 $h_1(x,t,p)=0$
event 11
event $1k_1$
derivative n $f_n(x,z,u,t,p)$
algebraic n $g_n(x,z,u,t,p)$
state event n $h_n(x,t,p)=0$
event n1
event $1k_n$

Figure 7: Simulator Structure for *Structural-Dynamic Systems.*

Figure 7 shows a structure for a simulator supporting structural dynamic modelling and simulation. The figure summarises the outlined ideas by extending the CSSL structure by ‘connection’ models, model-changing state events and multiple models. The main extension is that the translator generates not only one DAE model; he generates several DAE models from the (sub)model descriptions, and external events from the connection model, controlling the model execution sequence in the highest level of the dynamic event list. There, all (sub)models may be precompiled, or the new recent state space may be determined and translated to a DAE system in case of the external event (interpretative technique).

Clearly, not only ODE solver can make use of the model descriptions (derivatives), but also eigenvalue analysis and steady state calculation may be used and other analysis algorithms. Furthermore, complex experiments can be controlled by external events scheduling the same model in a loop. A simulator structure as proposed in Figure 7 is a very general one, because it allows as well external as ell as internal events, so that hybrid coupling with variable state models of any kind is possible.

Both approaches have advantages and disadvantages. The classical Dymola approach generates a fast simulation, because of the monolithic program. However, the state space is static. A hybrid approach handles separate model parts and must control the external events. Consequently, two levels of programs have to be generated: dynamic models, and a control program – today’s implementations are interpretative and not compiling, so that simulation times increase - but the overall state space is indeed dynamic.

A challenge for the future lies in the combination of both approaches. The main ideas are:

- Moderate hybrid decomposition
- Efficient implementation of models /control

For instance, for parameter state events (**SE-P**) an implementation within a model may be sufficient, for an event of **SE-S** type implementation with a model change may be advantageous because of easier state re-initialisation, and for a structural model change (**SE-D**) an implementation with hybrid decomposition may be preferred, because of much easier handling of the dynamic state change – and less necessity for index reduction. An efficient control of the sequence of models can be made by state charts, but also by a well-defined definitions and distinction of IF - and WHEN - constructs, like discussed in extensions of Scilab/Scicos for Modelica models.

3.4 Classification of Structural Features

While the *Classical Features* address the CSSL-standard and its extensions, *Structural Features* characterise features for physical modelling and for structural dynamic systems, the main development from the year 2000 on. The *Structural Features* may be classified as follows:

- Support of a-causal physical modelling at textual (PM-T) or graphical level (PM-G),
- Modelica standard (MOD) for physical modelling,
- Decomposition of structural dynamic systems with dynamic features (SD)
- Support of state chart modelling or of a similar construct, by means of textual (SC-T) or graphical (SC-G) constructs.

Simulators with a-causal modelling may support hybrid decomposition or not, and state chart modelling may be available or not. Simulators with features for state chart modelling may support hybrid decomposition or not, and a-causal modelling may be offered or not. In general, interpreter-oriented simulators offer more structural flexibility, but modern software structures would allow also flexibility with precompiled models.

In addition, of interest are also structural features as

- simulation-driven visualisation (visualisation objects defined with model objects; VIS),
- frequency domain analysis and linearization for steady state analysis (FA), and

- extended environment for complex experiments and data processing (ENV).

Appendix A summarises the availability of these *Structural Features* in some frequently used simulators, and especially in simulators understanding the MODELICA modelling notation (Section 3.1), together with the classical features. Basis for the classification are solutions to specific ARGESIM Benchmarks: Table 5 documents the evaluation of the benchmarks wrt to classical – 'C' -and structural – 'S' - features.

Table 5 also list features for further evaluation:

- optimisation and identification (OPT/ID)
- VHDL-AMS standard (V-AMS)
- System Dynamics modelling (SYS-D)
- Real-time Simulation (RT)
- Co-Simulation (COS)
- Spatial Dynamics (SPAT)
- Parallel Simulation (PAR)

4. IMPLEMENTATION EXAMPLES

As example, structural dynamic implementations of the *Constrained Pendulum* within the experimental simulator *Mosilab* are shown. Since 2004, Fraunhofer Gesellschaft Dresden develops a generic simulator *Mosilab*, which also initiates an extension to Modelica: multiple models controlled by state automata, coupled in serial and in parallel. Furthermore, Mosilab puts emphasis on co-simulation and simulator coupling, whereby for interfacing the same constructs are used than for hybrid decomposition.

Table 5: Evaluation of Classical and Structural Features in specific Benchmarks

Feature	Type	Benchmarks
MS	C	C1, C3, C5, C7, C9, C11, C13
ED	C	C3, C5, C7, C9, C11, C12, C13, C18
TEH	C	C3, C9, C12, C13, C18
SEH	C	C5, C7, C11, C12, C13
DAE	C	C7, C11, C13
IR	C	C11, C13
PM-T	S	C1, C3, C5, C7, C9, C11, C13, C15, C19
PM-G	S	C1, C3, C5, C7, C9, C11, C13, C15, C19
VIS	S	C1, C3, C7, C9, C11, C12, C17
MOD	S	C1, C3, C5, C7, C9, C11, C13, C15
SC-T	S	C3, C5, C7, C9, C11, C12, C13
SC-G	S	C3, C5, C7, C9, C11, C12, C13
SD	S	C5, C7, C9, C11, C13, C15
FA	S	C1, C3, C11, C13
ENV	S	C1, C3, C5, C7, C9, C11, C12, C13, C15, C17, C18, C19
OPT/ID	(S)	C7, C15, C17, C18
V-AMS	(S)	C3, C5, C7, C9, C13
SYS-D	(C)	C1, C7, C15, C17
RT	(C)	C3, C9, C13, C18
COS	(C)	C9, C11, C18
SPAT	(C)	C17, C19
PAR	(S)	CP-1, C19

Mosilab is a generic Modelica simulator, so all classical features are met (ED, SEH, DAE, PM-T, and PM-G 'yes', and MOD '(yes)' – because of subset implementation at present, 2008). For DAE solving, variants of IDA-DASSL solver are used. Mosilab implements extended state chart modelling, which may be translated directly due to Modelica standard into equivalent IF – THEN constructs, or which can control different models and model executions (SC-T, SC-G, and SD 'yes'). At state chart level, state events of type SE-D control the switching between different models and service the events (E-SE-D). State events affecting a state variable (SE-S type) can be modelled at this external level (E-SE-S type), or also as classic internal event (I-SE-S). Mosilab translates each model separately, and generates a main simulation program out of state charts, controlling the call of the precompiled models and passing data between the models, so that the software model of Mosilab follows the structure in Figure 7.

Mosilab is in developing, so it supports only a subset of Modelica, and index reduction has not been implemented yet, so that MOD gets a '(yes)' in parenthesis, and IR gets a '(no)' – indicating that the feature is not available at present (2008), but is scheduled for the future. Index reduction at present not available in Mosilab, but planned (IR '(no)') - has become topic of discussion: case studies show, that hybrid decomposition of structural dynamic systems results mainly in DAE systems of index $n = 1$, so that index reduction may be bypassed (except models with contact problems).

Mosilab allows very different approaches for modelling and simulation tasks, to be discussed with the *Constrained Pendulum* example.

Table 6: Mosilab Model for *Constrained Pendulum* – State Chart Switching between Different Models

```
model Long
equation
 mass*vdot/l1 + mass*g*sin(phi)+damping*v = 0;
end Long;
model Short
equation
 mass*vdot/ls + mass*g*sin(phi)+damping*v = 0;
end Short;
event discrete Boolean lengthen(start=true),
 shorten(start = false);
equation
 lengthen =
 (phi>phipin);shorten=(phi<=phipin);
statechart
state ChangePendulum extends State;
 State Short,Long,startState(isInitial=true);
transition Long->Short event shorten action
end transition;
transition Short -> Long event lengthen
 action
end transition; end ChangePendulum;
```

In a *Mosilab Standard Modelica Model* the *Constrained Pendulum* is defined in the MOSILAB equation layer as implicit law or with graphical blocks as in Dymola (Table 4, Figure 4). In an *Mosilab Model with State Charts,* state charts may be used instead of IF - or WHEN - clauses, with much higher flexibility and readability in case of complex conditions. In a *Structural Dynamic Mosilab Model* state charts may switch externally between two different pendulum models, controlled externally by a state chart.

Clearly, in case of this simple model, different models would not be necessary. Here, the system is decomposed into two different models, `Short` pendulum model, and `Long` pendulum model (Table 6), switched by external state charts.

REFERENCES

Breitenecker F., Troch I., 2004. Simulation Software – Development and Trends. In: Unbehauen H., Troch I., Breitenecker F., eds. *Modelling and Simulation of Dynamic Systems / Control Systems, Robotics, and Automation*. Oxford: Eolss Publishers, .

Cellier, F.E., *1991. Continuous System Modeling.* New York, Springer,

Cellier, F.E., and E. Kofman. *2006, Continuous System Simulation*. New York, Springer

Fritzson, P, 2005. *Principles of Object-Oriented Modeling and Simulation with Modelica*. Wiley IEEE Press.

Nytsch-Geusen C, Schwarz P, 2005. MOSILAB: Development of a Modelica based generic simulation tool supporting model structural dynamics. *Proc. 4th Intern. Modelica Conference,* 527-535. March 2005, Hamburg.

Strauss J. C. 1967. The SCi continuous system simulation language (CSSL). *Simulation* 9: 281-303.

APPENDIX

Appendix A

Availability of Structural and Classical Features in Simulators and Simulation Systems

	MS - Model Sorting	ED -Event Description	THE – Time Event Handling	SEH -State Event Handling	DAE - DAE Solver	IR - Index Reduction	PM-T - Physical Modelling -Text	PM-G - Physical Modelling -Graphics	VIS – 'Onlie' - Visualisation	MOD – Modelica Modelling	SC-T – State Chart – Modelling - Text	SC-G – State Chart Modelling - Graphics	SD – Structural Dynamic Systems	FA – Frequency Analysis	ENV – Extended Environment
MATLAB	no	no	no	(yes)	(yes)	no	no	no	(yes)	no	no	no	yes	yes	yes
Simulink	yes	(yes)	(yes)	(yes)	(yes)	no	no	(no)	(yes)	no	no	no	no	yes	(yes)
MATLAB/Simulink	yes	yes	yes	yes	(yes)	no	no	(no)	(yes)	no	no	no	yes	yes	yes
Simulink/ Stateflow	yes	yes	yes	yes	(yes)	no	no	(no)	(yes)	no	(yes)	yes	no	yes	(yes)
ACSL	yes	yes	yes	yes	yes	no	no	(no)	(yes)	no	no	no	no	yes	yes
Dymola	yes	yes	yes	yes	yes	yes	yes	yes	yes	yes	(yes)	(yes)	no	(no)	(yes)
MathModelica	yes	yes	yes	yes	yes	yes	yes	yes	(yes)	yes	(no)	(yes)	no	(no)	(no)
MathModelica / Mathematica	yes	yes	yes	yes	yes	yes	yes	yes	yes	yes	(no)	(yes)	yes	yes	yes
Mosilab	yes	yes	yes	yes	yes	(no)	yes	yes	(no)	(yes)	yes	yes	yes	no	(yes)
Open Modelica	yes	yes	yes	yes	yes	yes	yes	(no)	(no)	yes	(no)	(yes)	no	no	no
SimulationX	yes	yes	yes	yes	yes	yes	yes	yes	yes	yes	(no)	(yes)	no	yes	(yes)
AnyLogic	yes	yes	yes	(yes)	(yes)	no	no	no	yes	no	yes	yes	yes	no	no
Model Vision	yes	yes	yes	yes	yes	yes	yes	no	yes	no	yes	yes	yes	yes	no
Scilab	no	no	no	(yes)	(yes)	no	no	no	(yes)	no	no	no	yes	yes	yes
Scicos	yes	(yes)	(yes)	yes	yes	(yes)	yes	yes	(yes)	(yes)	yes	(yes)	no	no	no
Scilab/ Scicos	yes	yes	yes	yes	yes	(yes)	yes	yes	(yes)	(yes)	yes	(yes)	yes	yes	yes
(MapleSim)	yes	(yes)	(no)	(yes)	yes	yes	yes	yes	yes	yes	no	no	(yes)	(yes)	yes

INTERNAL/EXTERNAL EVENT – STRUCTURE IN SIMULATORS – CASE STUDIES WITH ARGESIM COMPARISONS

Felix Breitenecker[(a)], Siegfried Wassertheurer[(b)], Štefan Emrich[(c)], Nikolas Popper[(d)], Günther Zauner[(e)]

[(a) (c) (e)]Vienna Univ. of Technology, Austria
[(b)]ARCS - smart Biomedical Systems, Austria
[(d) (e)]“die Drahtwarenhandlung” Simulation Services, Austria

[(a)]Felix.Breitenecker@tuwien.ac.at, [(b)]siegfried.wassertheurer@arcsmed.at, [(c)]semrich@aurora.anum.tuwien.ac.at,
[(d)]niki.popper@drahtwarenhandlung.at, [(e)]Guether.Zauner@drahtwarenhandlung.at,

ABSTRACT

Object-oriented approaches, UML – notation, DAE modelling, variable structure modelling, Modelica notation and other developments have extended the CSSL standard for simulation languages essentially. After a review of the extended CSSL structure, this contribution classifies state events in continuous system modelling and develops a concept of internal / external state events, which allows modelling of structural dynamic systems in a proper way. Additionally, this concept can be ‘mapped’ onto the structure of several simulation languages. The application of the concept is documented by several implementation examples.

Keywords: Simulation software, hybrid structures, structural dynamic systems, feature comparison

1. INTRODUCTION – CLASSIC CSSL STRUCTURE

Simulation supported various developments in engineering and other areas, and simulation groups and societies were founded. One main effort of such groups was to standardise digital simulation programs and to work with a new basis: not any longer simulating the analog computer, but a self-standing structure for simulation systems. There were some unsuccessful attempts, but in 1968, the CSSL Standard became the milestone in the development: it unified the concepts and language structures of the available simulation programs, it defined a structure for the model, and it describes minimal features for a runtime environment.

The CSSL standard suggests structures and features for a model frame and for an experimental frame. This distinction is based on Zeigler’s concept of a strict separation of these two frames. Model frame and experimental frame are the user interfaces for the heart of the simulation system, for the simulator kernel or simulation engine. A translator maps the model description of the model frame into state space notation, which is used by the simulation engine solving the system governing ODEs. This basic structure of a simulator is illustrated in Figure 1; an extended structure with service of discrete elements is given in figure 3.

In CSSL’s model frame, a system can be described in three different ways, as an interconnection of blocks, by mathematical expressions, and by conventional programming constructs as in FORTRAN or C.

Mathematical basis is for the simulation engine is the state space description

$$\dot{\vec{x}}(t) = \vec{f}(\vec{x}(t), \vec{u}(t), t, \vec{p}), \quad \vec{x}(t_0) = \vec{x}_0\,, \tag{1}$$

which is used by the ODE solvers of the simulation engine. Any kind of textual model formulation, of graphical blocks or structured mathematical description or host languages constructs must be transformed to an internal state equation of the structure given above, so that the vector of derivatives $\vec{f}(\vec{x}, \vec{u}, t, \vec{p})$ can be calculated for a certain time instant $\vec{f}_i = \vec{f}_i(\vec{x}(t_i), \vec{u}(t_i), t_i, \vec{p})$. This vector of derivates is fed into an ODE solver in order to calculate a state update $\vec{x}_{i+1} = \Phi(\vec{x}_i . \vec{f}_i, h)$, h stepsize (all controlled by the simulation engine).

Essential is CSSL’s concept of SECTIONs or REGIONs, giving a certain structure to the model description. First, CSSL defines a set of operators like INTEG, which formulates parts of the state space description for the system governing ODEs. Other memory operators like DELAY for time delays, TABLE functions for generating (technical) tables, and transfer functions complete dynamic modelling parts. The dynamic model description builds up the DYNAMIC or DERIVATIVE section of the model description. Mapping the model description onto state space description, requires automatic sorting of the equations (blocks) to proper order of the calculation – an essential feature of the translator.

Sometimes together with the state space equations we also meet parameter equations, parameter dependent initial values, and calculations with the terminal values (e.g. for cost functions in an optimisation). In principle, all this calculations could be done in the dynamic model description, but then they are calculated at each evaluation of the derivative vector of the ODE solver – although they have to be calculated only once.

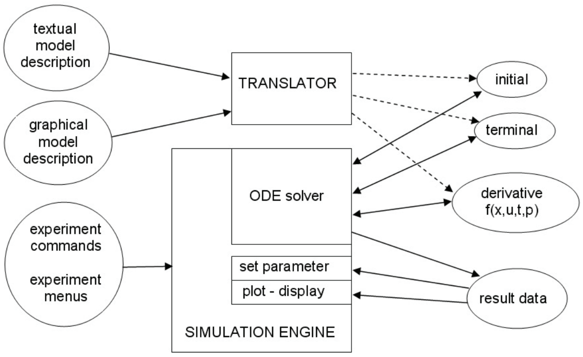


Figure 1: Basic Structure of a Simulation Language due to CSSL Standard

As example, we consider the model description for a pendulum. The well-known equations (length l, mass m, and damping coefficient d) and initial values and parameters are given by

$$\begin{aligned} &\ddot{\varphi}(t) = -\frac{g}{l}\sin\varphi - \frac{d}{m}\dot{\varphi}, \\ &\varphi_0 = \frac{\pi}{n},\ \dot{\varphi}_0 = 0, a = \frac{g}{l}, b = \frac{d}{m} \end{aligned} \tag{2}$$

A structured model description in ACSL (Section 5.2) generates efficient code: only the DERIVATIVE section is translated into the derivative vector function, while INITIAL and TERMINAL section are translated into functions called evaluated separately only once (Table 1).

It is task of the translator, to recognise the static elements, and to sort them separately from the dynamic equations, so that for the simulation engine dynamic equations (derivative), initial and parameter equations (initial), and terminal equations (terminal), are provided in separate modules.

Table 1: ACSL Structured Textual Model Description

```
PROGRAM math_pendulum
! --- structured CSSL model --------------------
! --- model parameters -------------------------
  CONSTANT m=1, l=1, d=0.3 ! kg, m, N*s/m
  CONSTANT g=9.81, pi=3.141592653; dphi0=0
  CONSTANT pintel=2
INITIAL ! calculation with parameters-----------
    phi0 = pi/pintel; a = g/l; b = d/m
END ! of INITIAL -------------------------------
  DERIVATIVE ! ODE model --------------------
    phi = integ ( dphi,        phi0)
    dphi = integ (-gdl*sin(phi)-ddm*dphi, dphi0)
  END ! of DERIVATIVE --------------------------
TERMINAL ! calculations with final states ------
    phi_grad = phi*180/pi
END ! of TERMINAL ------------------------------
END ! of Program -------------------------------
```

2. DISCRETE ELEMENTS IN CONTINUOUS SIMULATION – ARGESIM BENCHMARKS

The CSSL standard also defines segments for discrete actions, first mainly used for modelling discrete control. So-called DISCRETE regions or sections manage the communication between discrete and continuous world and compute the discrete model parts.

In graphical model description, discrete controllers and the time delay could be modelled by a **z-transfer** blocks, delay blocks and discrete controllers.

New versions of e.g. SIMULINK (Section 5.1) offer more complex discrete model parts, as triggered submodels, which can be executed only at one time instant.

For incorporating discrete actions, the simulation engine must interrupt the ODE solver and handle the event. For generality, efficient implementations set up and handle event lists, representing the time instants of discrete actions and the calculations associated with the action, where in-between consecutive discrete actions the ODE solver is to be called.

In order to incorporate DAEs and discrete elements, the simulator's translator must now extract from the model description the dynamic differential equations (derivative), the dynamic algebraic equations (algebraic), and the events (event i) with static algebraic equations and event time, as given in Figure 2 (extended structure of a simulation language due to CSSL standard). In principle, initial equations, parameter equations and terminal equations (initial, terminal) are special cases of events at time t = 0 and terminal time. Some simulators make use of a modified structure, which puts all discrete actions into one event module, where CASE - constructs distinguish between the different events.

2.1. State Events in Continuous Models

Much more complicated, but defined in CSSL, are the so-called state events. Here, a discrete action takes place at a time instant, which is not known in advance, it is only known as a function of the states.

As example, we consider the pendulum with constraints - *Constrained Pendulum*, being one of the so-called ***ARGESIM Benchmarks for Modelling and Simulation*** (published in the journal **SNE – Simulation News Europe**). If the pendulum is swinging, it may hit a pin positioned at angle φ_p with distance l_p from the point of suspension. In this case, the pendulum swings on with the position of the pin as the point of rotation. The shortened length is $l_s = l - l_p$. and the angular velocity $\dot{\varphi}$ is changed at position φ_p from $\dot{\varphi}$ to $\dot{\varphi}\cdot l / l_s$, etc. These discontinuous changes are state events, not known in advance.

For such events, the classical state space description is extended by the so-called state event function $h(\vec{x}(t),\vec{u}(t),\vec{p})$, the zero of which determines the event:

$$\begin{aligned} &\vec{x}(t) = \vec{f}(\vec{x}(t),\vec{u}(t),\vec{p},t), \\ &h(\vec{x}(t),\vec{u}(t),\vec{p},t) = 0 \end{aligned} \tag{3}$$

The event actions are usually discontinuous changes of parameters, inputs, states and model equations or model dimension, resp. In this notation, the model for *Constrained Pendulum* is given by

$$\dot{\varphi}_1 = \varphi_2,\ \dot{\varphi}_2 = -\frac{g}{l}\sin\varphi_1 - \frac{d}{m}\varphi_2, \qquad (4)$$
$$h(\varphi_1, \varphi_2) = \varphi_1 - \varphi_p = 0$$

The example involves two different events: change of length parameter, and change of state, i.e. angle velocity. Generally, state events (**SE**) can be classified in four types:

- **Type 1** – parameters change discontinuously (SE-P),
- **Type 2** - inputs change discontinuously (SE-I),
- **Type 3** - states change discontinuously (SE-S), and
- **Type 4** - state vector dimension changes (SE-D), including total change of model equations.

The event actions are discontinuous changes of parameters, inputs, states and model equations or model dimension, resp. Suppose, t^* is the (unknown) time instant of the event, the changes may be given in mathematical notation, where t^{*-} indicates the value before, and t^{*+} the values after the event:

$$\begin{aligned} &p(t^{*-}) \rightarrow p(t^{*+}) \\ &\vec{u}(t^{*-}) \rightarrow \vec{u}(t^{*+}) \\ &\vec{x}(t^{*-}) \rightarrow \vec{x}(t^{*+}) \\ &n(t^{*-}) \rightarrow n(t^{*+}) \end{aligned} \qquad (5)$$

In case of the *Constrained Pendulum* the events are hitting and leaving the pin, and the discontinuous changes are given by

$$\begin{aligned} &l(t^{hit^-}) \rightarrow l(t^{hit^{-+}}), \quad l(t^{hit^{-+}}) = l_s \\ &l(t^{leave^-}) \rightarrow l(t^{leave^{-+}}), l(t^{leave^{-+}}) = l \\ &\dot{\varphi}(t^{hit^-}) \rightarrow \dot{\varphi}(t^{hit^{-+}}), \\ &\qquad \dot{\varphi}(t^{hit^{-+}}) = \frac{l_s}{l}\dot{\varphi}(t^{hit^-}) \\ &\dot{\varphi}(t^{leave^-}) \rightarrow \dot{\varphi}(t^{leave^{-+}}), \\ &\qquad \dot{\varphi}(t^{leave^{-+}}) = \frac{l}{l_s}\dot{\varphi}(t^{leave^-}) \end{aligned} \qquad (6)$$

State events type 1 (**SE-P**) could also be formulated by means of IF-THEN-ELSE constructs and by switches in graphical model descriptions, without synchronisation with the ODE solver. The necessity of a state event formulation depends on the accuracy wanted. Big changes in parameters may cause problems for ODE solvers with step size control.

State events of type 3 (**SE-S**) are essential state events. They must be located, transformed into a time event, and modelled in discrete model parts.

State events of type 4 (**SE-D**) are also essential ones. In principle, they are associated with hybrid modelling: models following each other in consecutive order build up a sequence of dynamic processes. And consequently, the structure of the model itself is dynamic.

2.2. State Event Handling

The handling of a state event requires four steps:

1. Detection of the event, usually by checking the change of the sign of h(x) within the solver step over [ti, ti+1]
2. Localization of the event by a proper algorithm determining the time t* when the event occurs and performing the last solver step over [ti, t*]
3. Service of the event: calculating / setting new parameters, inputs and states; switching to new equations
4. Restart of the ODE solver at time t* with solver step over [t*= ti+1, ti+2]

State events are facing simulators with severe problems. Up to now, the simulation engine had to call independent algorithms, now a root finder for the state event function *h* needs results from the ODE solver, and the ODE solver calls the root finder by checking the sign of *h*. For finding the root of the state event function *h(x)*, either interpolative algorithms (MATLAB/Simulink; Section 5.1) or iterative algorithms are used (ACSL, Section 5.2; Dymola, Section 5.3.1)

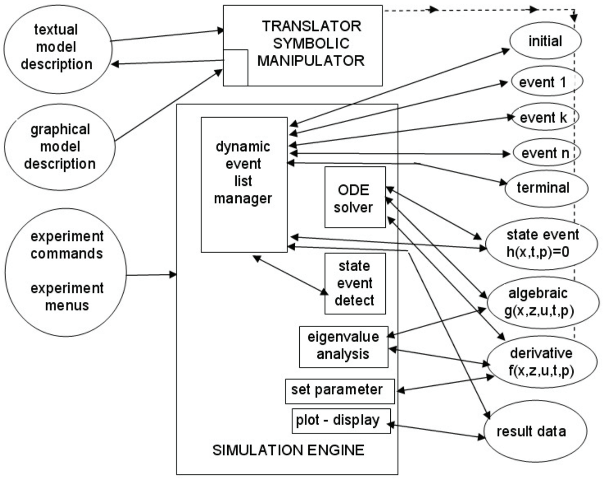


Figure 2: Extended Structure of a Simulation System due to Extensions of the CSSL Standard with Discrete Elements and with DAE Models.

Figure 2 also shows the necessary extensions for incorporating state events. The simulator's translator must extract from the model description additionally the state event functions (state event j) with the associated event action – only one state event shown in the figure). In the simulator kernel, the static event management must be made dynamically: state events are dynamically handled and transformed to time events. In principle, the kernel of the simulation engine has become an event handler, managing a complex event list with feedbacks.

In case of a structural change of the system equations (state event of type 4 – **SE-D**), simulators usually can manage only fixed structures of the state space. The technique used is to 'freeze' the states that are bound by conditions causing the event. In case of a complete change of equations, both systems are calculated together, freezing one according to the event.

One way around is to make use of the experimental frame: the simulation engine only detects and localises the event, and updates the system until the event time. Then control is given back to the experimental frame. The state event is now serviced in the experimental frame, using features of the environment. Then a new simulation run is restarted (modelling of the structural changes in the experimental frame).

Table 2: *Constrained Pendulum*: Continuous Model with State Events (ACSL)

```
PROGRAM constrained pendulum
CONSTANT m = 1.02, g = 9.81, d =0.2
CONSTANT lf=1, lp=0.7
DERIVATIVE dynamics
  ddphi = -g*sin(phi)/l - d*dphi/m
  dphi  = integ ( ddphi, dphi0)
  phi   = integ ( dphi, phi0)
  SCHEDULE hit   .XN. (phi-phip)
  SCHEDULE leave .XP. (phi-phip)
END ! of dynamics
DISCRETE hit
  l = ls; dphi = dphi*lf/ls
END ! of hit
DISCRETE leave
  l = lf; dphi = dphi*ls/lf
END ! of leave;
END ! of constrained pendulum
```

The *Constrained Pendulum* example involves a state event of type 1 (**SE-P**) and type 3 (**SE-S**). A classical ACSL (Section 5.2) model description works with two discrete sections `hit` and `leave`, representing the two different modes, both called from the dynamic equations in the derivative section (Table 2). Dymola (Section 5.3.1) defines events and their scheduling implicitly by WHEN – or IF - constructs in the dynamic model description, in case of the discussed example e.g. by

```
WHEN phi-phip=0 AND phi>phip
THEN l = ls; dphi = dphi*lf/ls
```

In case of more complex event descriptions, the WHEN – or IF – clauses are put into an ALGORITHM section, similar to ACSL's DISCRETE section.

In graphical model descriptions, we are faced with the problem that calculations at discrete time instants are difficult to formulate. For the detection of the event, SIMULINK provides the `HIT CROSSING` block (in new Simulink version implicitly defined). This block starts state event detection (interpolation method) depending on the input, the state event function, and outputs a trigger signal, which may call a triggered subsystem servicing the event.

2.3. ARGESIM Benchmarks

In 1990, the journal ***SNE – Simulation News Europe*** – started a series on *Comparison of Simulation Software*, which has been developed to *Benchmarks for Modelling and Simulation Techniques*. Up to now, 20 comparisons and benchmarks have been defined, and about 250 solutions have been published – being a very valuable source for discussing and documenting various aspects of modelling and simulation approaches.

Some of these benchmarks address state events, hybrid systems, and structural dynamic systems:

- C 3 - Analysis of a Generalized Class-E Amplifier
- C 5 - Two State Model
- C 7 - Constrained Pendulum
- C 11 - SCARA Robot
- C 12 - Collision Processes in Rows of Spheres
- C 13 - Crane Crab with Embedded Control

This contribution mainly concentrates on Benchmark C5 *Constrained Pendulum*, involving state events of type **SE-P** and **SE-S**. With respect to state event types, the following list gives information about occurrence or possible model approaches in benchmarks:

- **SE-P**: C3, C5, C7, C11, C12, C13
- **SE-T**: C3, C12, C13
- **SE-S**: C3, C5, C7, C11, C12, C13
- **SD-D**: C3, C5, C7, C11, C13

At present, further benchmarks are in preparation, among them an extended benchmark for hybrid systems. Detailed information about definitions and solutions to these benchmarks can be found in SNE, www.argesim.org.

3. MODELLING WITH STATE CHARTS

In the end of the 1990s, computer science initiated a new development for modelling discontinuous changes. The *Unified Modelling Language* (UML) is one of the most important standards for specification and design of object oriented systems. This standard was tuned for real time applications in the form of a new proposal, *UML Real-Time* (UML-RT). By means of UML-RT, objects can hold the dynamic behaviour of an ODE.

In 1999, a simulation research group at the Technical University of St. Petersburg used this approach in combination with a hybrid state machine for the development of a hybrid simulator *AnyLogic* (Section 5.4). The modelling language is an extension of UML-RT; the main building block is the *Active Object*. Active objects have internal structure and behaviour, and allow encapsulating of other objects to any desired depth. Relationships between active objects set up the hybrid model.

Active objects interact with their surroundings solely through boundary objects: ports for discrete communication, and variables for continuous communication (Figure 3). The activities within an object are usually defined by state charts (extended state machine).

While discrete model parts are described by means of state charts, events, timers and messages, the continuous model parts are described by means of ODEs and DAEs in CSSL-type notation and with state charts within an object.

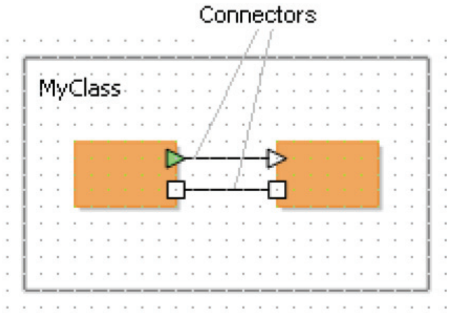
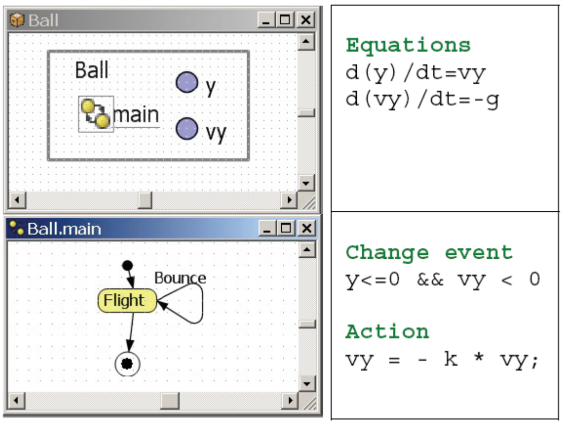


Figure3: Active Objects with Connectors - Discrete Messages (Rectangles) and Continuous Signals (Triangles)

An AnyLogic implementation of the well-known *Bouncing Ball* example shows a simple use of state chart modelling (Figure 4). The model equations are defined in the active object ball, together with the state chart ball.main. This state chart describes the interruption of the state flight (without any equations) by the event bounce (**SE-P** and **SE-S** event) defined by condition and action.

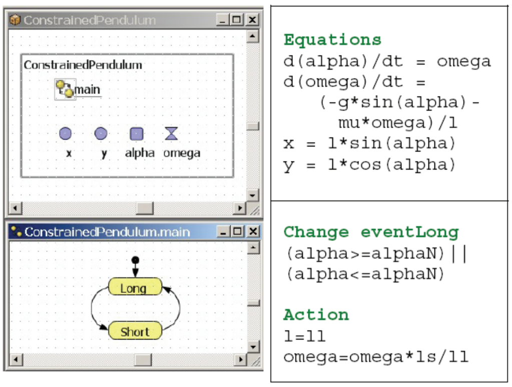


```
Equations
d(y)/dt=vy
d(vy)/dt=-g
```

```
Change event
y<=0 && vy < 0

Action
vy = - k * vy;
```

Figure 4: AnyLogic Model for the *Bouncing Ball*

AnyLogic influenced further developments for hybrid and structural dynamic systems, and led to a discussion in the Modelica community with respect to a proper implementation of state charts in Modelica. The principle question is, whether state charts are to be seen as comfortable way to describe complex WHEN – and IF – constructs, being part of the model, or whether state charts control different models from a higher level. At present (2008) a free Modelica state chart library 'emulates' state charts by Boolean variables and IF – THEN – ELSE constructs. A further problem is the fact, that the state chart notation is not really standardised; AnyLogic makes use of the Harel state chart type.



```
Equations
d(alpha)/dt = omega
d(omega)/dt =
    (-g*sin(alpha)-
     mu*omega)/l
x = l*sin(alpha)
y = l*cos(alpha)
```

```
Change eventLong
(alpha>=alphaN)||
(alpha<=alphaN)

Action
l=ll
omega=omega*ls/ll
```

Figure 5: AnyLogic model for *Constrained Pendulum*, Simple Implementation

4. HYBRID AND STRUCTURAL-DYNAMIC SYSTEMS

Hybrid systems often come together with a change of the dimension of the state space, then called *structural-dynamic systems*. The dynamic change of the state space is caused by a state event of type **SE-D**. In contrary to state events **SE-P** and **SE-S**, states and derivatives may change continuously and differentiable in case of structural change. In principle, *structural-dynamic systems* can be seen from two extreme viewpoints. The one says, in a maximal state space, state events switch on and off algebraic conditions, which freeze certain states for certain periods. The other one says that a global discrete state space controls local models with fixed state spaces, whereby the local models may be also discrete or static. These viewpoints derive two different approaches, the *maximal state space*, and the *hybrid decomposition*.

4.1. Maximal State Space for Structural-Dynamic Systems – Internal Events

Most implementations of physically based model descriptions support a big monolithic model description, derived from laws, ODEs, DAEs, state event functions and *internal events*. The state space is maximal and static, index reduction in combination with constraints keep a consistent state space. The approach can be classified with respect to event implementation. The approach handles all events of any kind (**SE-P**, **SE-S**, and **SE-D**) within the ODE solver frame, also events which change the state space dimension (change of degree of freedoms) – consequently called *internal events* – **I-SE**.

Using the classical state chart notation, internal state events I-SE caused by the model schedule the model itself, with usually different re-initialisations (depending on the event type I-SE-P, I-SES, I-SE-D; Fig. 6). VHDL-AMS and Dymola follow this approach, handling also DAE models with index > 1. Discrete model parts are only supported at event level. ACSL and MATLAB / Simulink generate also a maximal state space.

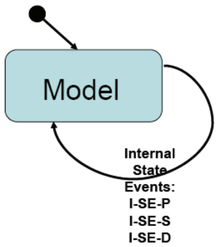


Figure 6: State Chart Control for *Internal Events* of one Model

4.2. Hybrid Decomposition for Structural-Dynamic Systems – External Events

The hybrid decomposition approach makes use of *external events* (**E-SE**), which controls the sequence and the serial coupling of one model or of more models. A convenient tool for switching between models is a state chart, driven by the *external events* – which itself are generated by the models. Control for continuous models and for discrete actions can by modelled by state charts. Figure 7 (left) shows the hybrid coupling of two models, which may be extended to an arbitrary number of models, with possible events **E-SE-P**, **E-SE-S**, and **ESE-D**.

As special case, this technique may be also used for serial conditional 'execution' of one model – Figure 7 (only for SE-P and SE-S).

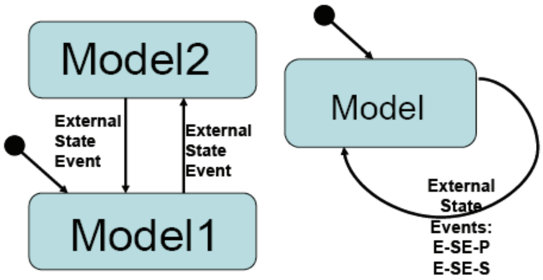


Figure 7: State Chart Control for *External Events* for two Models (left) and for one Model (right).

This approach additionally allows not only dynamically changing state spaces, but also different model types, like ODEs, linear ODEs (to be analysed by linear theory), PDEs, etc. to be processed in serial or also in parallel, so that also co-simulation can be formulated based on external events. The approach allows handling all events also outside the ODE solver frame. After an event, a very new model can be started. This procedure may make sense especially in case of events of type **SE-D** and **SE-S**. As consequence, consecutive models of different state spaces may be used.

Figure 8 shows a structure for a simulator supporting structural dynamic modelling and simulation. The figure summarises the outlined ideas by extending the CSSL structure by control model, external events and multiple models. The main extension is that the translator generates not only one DAE model; he generates several DAE models from the (sub)model descriptions, and external events from the connection model, controlling the model execution sequence in the highest level of the dynamic event list.

There, all (sub-) models may be precompiled, or the new recent state space may be determined and translated to a DAE system in case of the external event (interpretative technique).

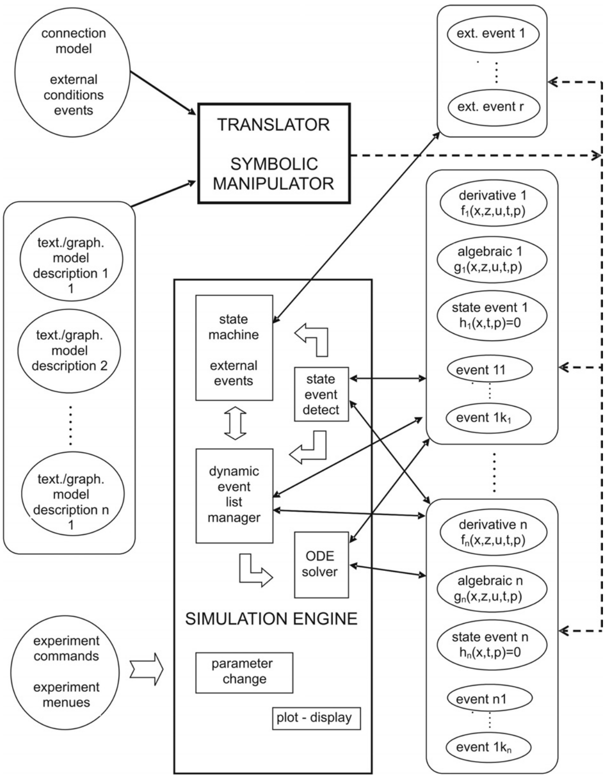

connection model
external conditions events
TRANSLATOR
SYMBOLIC MANIPULATOR
ext. event 1
ext. event r
text./graph. model description 1 1
text./graph. model description 2
text./graph. model description n 1
state machine
external events
state event detect
dynamic event list manager
ODE solver
SIMULATION ENGINE
parameter change
plot - display
experiment commands
experiment menues
derivative 1 $f_1(x,z,u,t,p)$
algebraic 1 $g_1(x,z,u,t,p)$
state event 1 $h_1(x,t,p)=0$
event 11
event $1k_1$
derivative n $f_n(x,z,u,t,p)$
algebraic n $g_n(x,z,u,t,p)$
state event n $h_n(x,t,p)=0$
event n1
event $1k_n$

Figure 8: Structure for a Simulation System with *External State Events* E-SE and Classical Internal State Events I-SE for Controlling Different Models.

4.3. Mixed Approach with Internal and External Events

A simulator structure as proposed in Figure 8 is a very general one, because it allows as well external as ell as internal events, so that hybrid coupling with variable state models of any kind with internal and external events is possible (Figure 9). Both approaches have advantages and disadvantages. The classical Dymola approach generates a fast simulation, because of the monolithic program. However, the state space is static. A hybrid approach handles separate model parts and must control the external events. Consequently, two levels of programs have to be generated: dynamic models, and a control program – today's implementations are interpretative and not compiling. A challenge for the future lies in the combination of both approaches. The main ideas are:

- Moderate hybrid decomposition
- External and internal events
- Efficient implementation of models and control

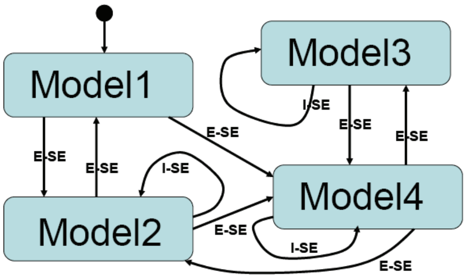


Figure 9: State Chart Control for Different Models with *Internal* and *External Events*.

For instance, for parameter state events (**SE-P**) an implementation with an internal event may be sufficient (**I-SE-P**), for an event of **SE-S** type implementation with an external event may be advantageous because of easier state re-initialisation (**E-SE-S**), and for a structural model change (**SE-D**) an implementation with an external event may be preferred (E-SE-D), because of much easier handling of the dynamic state change – and less necessity for index reduction. An efficient control of the sequence of models can be made by state charts, but also by a well-defined definitions and distinction of IF - and WHEN - constructs, like discussed in extensions of Scilab/Scicos for Modelica models.

5. STRUCTURAL FEATURES IN SIMULATORS

Structural dynamic system are up to now – 2008 – a challenge for simulators. In principle, model-compiling simulators must 'emulate' the dynamic structure in a maximal state space by switching between 'active' states, while interpreting simulators can switch between different models by means of a control model handling the structural changes.

But there exist also mixed strategies. In the following, some simulators are discussed with respect to their features for structural dynamic systems. Mainly using the benchmark Constrained Pendulum, in detail features for state chart modelling (as convenient tool for control models) and features for hybrid decomposition are investigated:

- Support of state chart modelling or of a similar construct, by means of textual or graphical constructs.
- Decomposition of structural dynamic systems with dynamic features– features for external events.

5.1. MATLAB / Simulink / Stateflow

The mainly interpretative systems MATLAB / Simulink offer different approaches. First, MATLAB itself allows any kind of static and dynamic decomposition, but MATLAB is not a simulator, because the model equations have to be provided in a sorted manner.

Second, MATLAB allows hybrid decomposition at MATLAB level with Simulink models. There, from MATLAB level, different Simulink models are called conditionally, and in Simulink, a state event is determined by the hit-crossing block (terminating the simulation). For control, in MATLAB only IF – THEN constructs are available. Table 3 – MATLAB control model, and Figure 10 – graphical Simulink model, show a hybrid decomposition of this type for the *Constrained Pendulum.*

Table 3: MATLAB Control Model for Constrained Pendulum with External Events Switching between Long and Short Pendulum

```
if ((phi_p-phi0)*phi_p<0 |
              (phi0==phi_p & phi_p*v>0))
   dphi0=v/ls;
   sim('pendulum_short',[t(length(t)),10]);
   v=dphi(length(dphi))*ls;
else
   dphi0=v/l;
   sim('pendulum_long',[t(length(t)),10]);
   v=dphi(length(dphi))*l;
end
```

MATLAB is a very powerful environment with various modules. Simulink is MATLAB's simulation module for block-oriented dynamic models (directed signal graphs), which can be combined with Stateflow, MATLAB's module for event-driven state changes described by state chart.

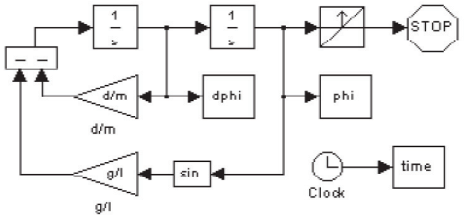


Figure 10: Simulink Model for *Constrained Pendulum* with *External Event* detected by Hit-Crossing Block.

At Simulink level, Stateflow, Simulink's state chart modelling tool, may control different submodels. These submodels may be dynamic models based on ODEs (DAEs), or static models describing discrete actions (events). Consequently, Stateflow can be used for implementation of the *Constrained Pendulum*, where the state charts control length and change of velocities in case of hit by triggering the static changes (Figure 11). A solely Simulink implementation would make use of a triggered submodels describing the events by AND – and OR – blocks, or by a MATLAB function.

Alternatively, for *Constrained Pendulum* Stateflow could control two different submodels representing long and short pendulum enabled and disabled by the state chart control. Internally Simulink generates a state space with 'double' dimension, because Simulink can only work with a maximal state space and does not allow hybrid decomposition.

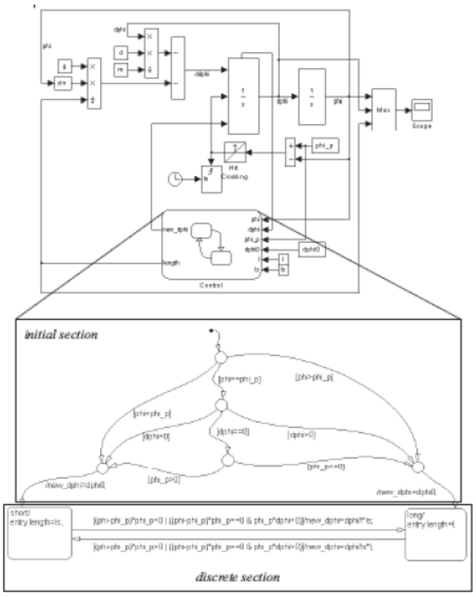


Figure11: Simulink Model for *Constrained Pendulum* with *External Event* (Hit-Crossing Block, Stateflow)

5.2. ACSL

ACSL – Advanced Continuous Simulation Language – has been developed since more than 25 years. ACSL was strongly influenced by the CSSL standard. ACSL' software structure is a direct mapping of the structure in Figure 2. Implementations of the *Constrained Pendulum* have been shown in the previous sections Table 1, Table 2), as example for modelling due to CSSL standard.

A very interesting additional ACSL module is an extended environment called ACSLMath. ACSLMath was intended to have same features as MATLAB; available is only a subset, but powerful enough for an extended environment, which can be used for hybrid decomposition of a structural dynamic model in almost the same way than MATLAB does (the MATLAB model in Table 3 can be used in ACSLMath, only by replacing the `sim` calls by `run` calls).

5.3. MODELICA - Simulators

In the 1990s, many attempts have been made to improve and to extend the CSSL structure, especially for the task of mathematical modelling. The basic problem was the state space description, which limited the construction of modular and flexible modelling libraries. Two developments helped to overcome this problem. On modelling level, the idea of physical modelling gave new input, and on implementation level, the object-oriented view helped to leave the constraints of input/output relations.

In physical modelling, a typical procedure for modelling is to cut a system into subsystems and to account for the behaviour at the interfaces. Balances of mass, energy and momentum and material equations model each subsystem. The complete model is obtained by combining the descriptions of the subsystems and the interfaces.

This approach requires a modelling paradigm different to classical input/output modelling. A model is considered as a constraint between system variables, which leads naturally to DAE descriptions. The approach is very convenient for building reusable model libraries.

These ideas stimulated the development of the simulator *Dymola*, whose modelling frame has been extended to a general standardised modelling language called *Modelica*. Modelica is intended for modelling within many application domains such as electrical circuits, multibody systems, drive trains, hydraulics, thermo-dynamical systems, and chemical processes etc. It supports several modelling formalisms: ordinary differential equations, differential-algebraic equations, bond graphs, finite state automata, and Petri nets etc. *Modelica* serves as a standard format so that models arising in different domains can be exchanged between tools and users.

Up to now – similar to VHDL-AMS – some simulation systems understand Modelica (2008; generic – new simulator with Modelica modelling, extension - Modelica modelling interface for existing simulator):

- Dymola from Dynasim (generic),
- MathModelica from MathCore Engineering (generic)
- SimulationX from ISI (generic/extension)
- Scilab/Scicos (extension)
- MapleSim (extension, announced)
- Open Modelica - since 2004 the University of Lyngby develops an provides an open Modelica simulation environment (generic),
- Mosilab - Fraunhofer Gesellschaft develops a generic Modelica simulator, which supports dynamic variable structures (generic)
- Dymola / Modelica blocks in Simulink

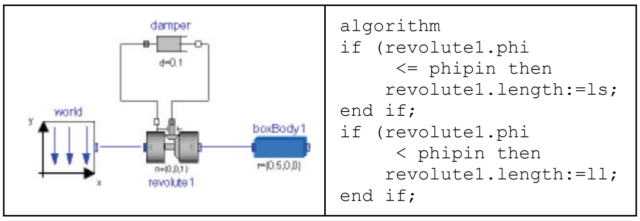
As Modelica also incorporates graphical model elements, the user may choose between textual modelling, graphical modelling, and modelling using elements from an application library. Furthermore, graphical and textual modelling may be mixed in various kinds. The minimal modelling environment is a text editor; a comfortable environment offers a graphical modelling editor.

The *Constrained Pendulum* example can be formulated in Modelica textually as a physical law for angular acceleration. The event with parameter change is put into an `algorithm` section, defining and scheduling the parameter event **SE-P** (Table 4). As instead of angular velocity, the tangential velocity is used as state variable, the second state event **SE-S** 'vanishes'.

Table 4: Textual Modelica Model for *Constrained Pendulum*

```
equation /*pendulum*/
  v = length*der(phi);
  vdot = der(v);
  mass*vdot/length + mass*g*sin(phi)
  +damping*v = 0;
algorithm
 if (phi<=phipin) then length:=ls; end if;
 if (phi>phipin) then length:=l1; end if;
```

But *Modelica* allows also combining textual and graphical modelling. For the *Constrained Pendulum* example, the basic physical dynamics could be modelled graphically with joint and mass elements, and the event of length change is described in an `algorithm` section, with variables interfacing to the predefined variables in the graphical model part (Figure 12).



```
algorithm
if (revolute1.phi
     <= phipin then
   revolute1.length:=ls;
end if;
if (revolute1.phi
     < phipin then
   revolute1.length:=l1;
end if;
```

Figure 12: Mixed Graphical and Textual Modelica Model for *Constrained Pendulum*

5.3.1. Dymola

Dymola was the first Modelica simulator. Dymola, introduced by F. E. Cellier as a-causal modelling language, and developed to a simulator by H. Elmquist, can be called the mother of Modelica. Dymola clearly can understand the Modelica models given in Table 4 and Figure 12. Dymola offers also a Modelica – compatible state chart library, which allows to model complex conditions (internally translated into IF – THEN – ELSE or WHEN constructs). Figure 14 shows an implementation of the *Constrained Pendulum* using this library.

5.3.2. Mosilab

Since 2004, Fraunhofer Gesellschaft Dresden develops a generic simulator *Mosilab*, which also initiates an extension to Modelica: multiple models controlled by state automata, coupled in serial and in parallel. Furthermore, Mosilab puts emphasis on co-simulation and simulator coupling, whereby for interfacing the same constructs are used than for hybrid decomposition. Mosilab is a generic Modelica simulator.

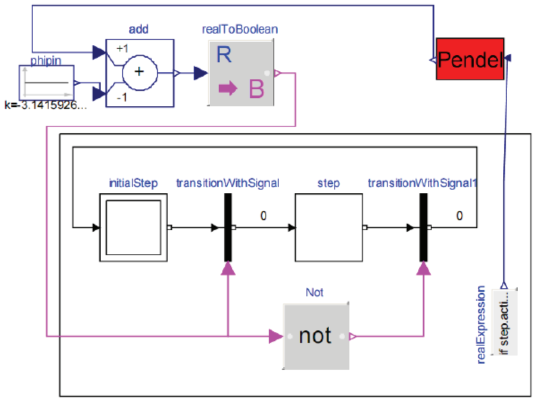


Figure 13: Graphical Dymola Model for *Constrained Pendulum* with *Internal Events* Managed by Elements of Dymola's State Chart Library

Mosilab implements extended state chart modelling, which may be translated directly due to Modelica standard into equivalent IF – THEN constructs, or which can control different models and model executions. At state chart level, state events of type **SE-D** control the switching between different models and service the events (**E-SE-D**). State events affecting a state variable (**SE-S** type) can be modelled at this external level (**E-SE-S** type), or also as classic internal event (**I-SE-S**). Mosilab translates each model separately, and generates a main simulation program out of state charts, controlling the call of the precompiled models and passing data between the models, so that the software model of Mosilab follows the structure in Figure 8. The constructs for the state charts are modifications of state chart modelling in AnyLogic.

Mosilab allows very different approaches for modelling and simulation tasks, to be discussed with the *Constrained Pendulum* example. Three different modelling approaches reflect the distinction between internal and external events as discussed before.

Mosilab Standard Modelica Model. In a standard Modelica approach, the *Constrained Pendulum* is defined in the MOSILAB equation layer as implicit law; the state event, which appears every time when the rope of the pendulum hits or 'leaves' the pin, is modelled in an `algorithm section` with if (or when) – conditions (Table 7).

Table 5: Mosilab Model for Constrained Pendulum – Standard Modelica Approach - *Internal Events* (I-SE-P)

```
equation /*pendulum*/
 v = l1*der(phi); vdot = der(v);
 mass*vdot/l1 + mass*g*sin(phi)+damping*v = 0;
algorithm
 if (phi<=phipin) then length:=ls; end if;
 if (phi>phipin) then length:=l1; end if;
end
```

Mosilab I-SE-P Model with State Charts. MOSILAB's state chart approach models discrete elements by state charts, which may be used instead of IF - or WHEN - clauses, with much higher flexibility and readability in case of complex conditions. There, Boolean variables define the status of the system and are managed by the state chart.

Table 6: Mosilab Model for *Constrained Pendulum* – State Chart Model with *Internal Events* (I-SE-P)

```
event Boolean lengthen(start=false),
 shorten(start = false);
equation
lengthen=(phi>phipin); shorten=(phi<=phipin);
equation /*pendulum*/
 v = l1*der(phi); vdot = der(v);
 mass*vdot/l1 + mass*g*sin(phi)+damping*v= 0;
statechart
 state LengthSwitch extends State;
 State Short,Long,Initial(isInitial=true);
transition Initial -> Long end transition;
transition Long -> Short event shorten action
 length := ls;
end transition;
….;
```

Table 6 shows a Mosilab implementation of the *Constrained Pendulum*: state charts initialise the system and manage switching between long and short pendulum, by changing the length appropriately.

Mosilab E-SE-P Model. Mosilab's state chart construct is not only a good alternative to IF - or WHEN - clauses within one model, it offers also the possibility to switch between structural different models. This very powerful feature allows any kind of hybrid composition of models with different state spaces and of different type (from ODEs to PDEs, etc.). Table 7 shows a Mosilab implementation of the *Constrained Pendulum* making use of two different pendulum models, controlled externally by a state chart.

Here, the system is decomposed into two different models, `Short` pendulum model, and `Long` pendulum model, controlled by a state chart. The model description (Table 7) defines now first the two pendulum models, and then the event as before. The state chart creates first instances of both pendulum models during the initial state (`new`). The transitions organise the switching between the pendulums (`remove`, `add`). The `connect` statements are used for mapping local to global state.

Table 7: Mosilab Model for *Constrained Pendulum* – State Chart Switching between Different Pendulums Models by *External Events* (E-SE-P)

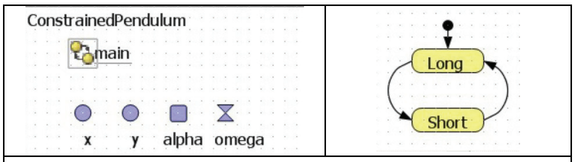
```
model Long
equation
 mass*vdot/l1 + mass*g*sin(phi)+damping*v = 0;
end Long;
model Short
equation
 mass*vdot/ls + mass*g*sin(phi)+damping*v = 0;
end Short;
event discrete Boolean lengthen(start=true),
equation
 lengthen =
 (phi>phipin);shorten=(phi<=phipin);
statechart
state ChangePendulum extends State;
 State Short,Long,startState(isInitial=true);
transition startState -> Long action
 L:=new Long(); K:=new Short(); add(L);
end transition;
transition Long->Short event shorten action
 disconnect ….; remove(L); add(K); connect …
end transition;
transition Short -> Long event lengthen
 action;disconnect…;remove(K);add(L);connect ……
end transition; end ChangePendulum;
```

5.4. AnyLogic – Hybrid State Chart Simulator

AnyLogic – already discussed in section 3) is based on hybrid automata. Consequently, hybrid decomposition and control by external events is possible. In AnyLogic, various implementations for the *Constrained Pendulum* are possible. A classical implementation is given in Figure 5, following classical textual ODE modelling, whereby instead of IF – THEN clauses a state chart is used for switching (**I-SE-P**, **I-SE-S**).

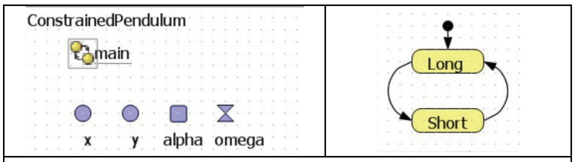
AnyLogic E-SE-P Model with State Charts. A hybrid decomposed model makes use of two different models, defined in substate / submodel `Short` and `Long`. – part of a state chart switching between these submodels. The events defined at the arcs stop the actual model, set new initials and start the alternative model (Figure 13).

AnyLogic E-SE-P Model with Parallel Models. AnyLogic works interpretatively, after each external event state equations are tracked and sorted anew for the new state space. This makes it possible, to decompose model not only in serial, but also in parallel (Figure 14).



```
Equations Constrained Pendulum
parameter …
end Constrained pendulum
Equations Short
d(alpha)/dt = omega
d(omega)/dt =(-g*sin(alpha)-mu*omega)/ls
Change eventLong;
(alpha>=alphaN)||(alpha<=alphaN)
Action; omega=omega*ls/ll; stop
end Short
Equations Long
d(alpha)/dt = omega
d(omega)/dt =(-g*sin(alpha)-mu*omega)/ll
Change eventShort
(alpha>=alphaN)||(alpha<=alphaN)
Action; omega=omega*ll/ls; stop
end Long
```

Figure 13: AnyLogic Model for *Constrained Pendulum*, Hybrid Model Decomposition with two Pendulum Models and *External Events*



```
Equations Constrained Pendulum
d(alpha)/dt = omega
x = l*sin(alpha); y = l*cos(alpha)
end Constrained pendulum
Equations Short
d(omega)/dt =(-g*sin(alpha)-mu*omega)/ls
Change eventLong
(alpha>=alphaN)||(alpha<=alphaN)
Action; omega=omega*ls/ll; stop
end Short
Equations Long
d(omega)/dt =(-g*sin(alpha)-mu*omega)/ll
Change eventShort
(alpha>=alphaN)||(alpha<=alphaN)
Action; omega=omega*ll/ls; stop
end Long
```

Figure 14: AnyLogic Model for *Constrained Pendulum*, Hybrid Model Decomposition with Two Models for Angular Velocity and Overall Model for Angle

REFERENCES

Breitenecker F., Troch I., 2004. Simulation Software – Development and Trends. In: Unbehauen H., Troch I., Breitenecker F., eds. *Modelling and Simulation of Dynamic Systems / Control Systems, Robotics, and Automation.* Oxford: Eolss Publishers, .

Fritzson, P, 2005. *Principles of Object-Oriented Modeling and Simulation with Modelica.* Wiley IEEE Press.

Nytsch-Geusen C, Schwarz P, 2005. MOSILAB: Development of a Modelica based generic simulation tool supporting model structural dynamics. *Proc. 4th Intern. Modelica Conference,* 527-535. March 2005, Hamburg.

Strauss J. C. 1967. The SCi continuous system simulation language (CSSL). *Simulation* 9: 281-303.

A PETRI NET APPROACH TO ARGESIM COMPARISON C2 ‘FLEXIBLE ASSEMBLY SYSTEM’ USING THE MATLAB PETRISIMM TOOLBOX

Thomas Löscher[(a)], **Felix Breitenecker**[(b)]

[(a)]PROFACTOR GmbH, Austria
[(b)] Vienna Univ. of Technology, Austria

[(a)]thomas.loescher@profactor.at, [(b)] felix.breitenecker@tuwien.ac.at

ABSTRACT

This paper deals with the modelling, simulation and optimisation of a flexible assembly system. Timed and Coloured Petri nets are used to represent the systems behaviour. The so called MATLAB PetriSimM toolbox forms the basis for modelling and extends the benchmark performed by the ARGESIM comparisons.

Keywords: process simulation, production simulation, resources planning

1. INTRODUCTION

Petri nets offer the capability to model discrete event systems on a very low level (Petri 1962). Representing resource sharing problems, conflicts or describing complex control strategies are the results of the use of their basic properties. A lot of applications exist in the field of modelling and simulating flexible manufacturing systems by means of Petri nets (Zuberek 1999). For evaluation purposes and management ratios it is necessary to simulate over the time domain. For Petri nets many extensions exist to implement time to their basic structure and definition (Zuberek 1991; Bowden 2000). The complexity of flexible manufacturing systems can be easily growing very high. Therefore other extensions are required to ensure the capability of meaningful modelling. Petri nets are also growing very fast if the complexity is increasing. Further extensions like the introduction of token colours are needed (Bause and Kritzinger 2002). In this simple approach the colours are used to simplify the graphical representation of the Petri net.

This paper shows the modelling, simulation and optimisation of a flexible assembly system. Basically, this problem definition is made for an entity flow approach where each processing unit is modelled as entity of the system. Petri nets do not really offer this way of modelling because tokens are destroyed if they are not needed any more and they are created if they are needed. Therefore it is not possible to track tokens as entities during the simulation and no further properties or attributes can be given to them. The use of Petri nets is a benchmark to test and show the feasibility of their capabilities.

2. PROBLEM DEFINITION

2.1. ARGESIM Comparisons

Simulation News Europe (SNE) features a series on comparisons of simulation software (Breitenecker 2008). Based on simple, easily comprehensible models special features of modelling and experimentation within simulation languages, also with respect to an application area, are compared.

Features are, for instance: modelling technique, event handling, numerical integration, steady-state calculation, distribution fitting, parameter sweep, output analysis, animation, complex logic strategies, sub-models, macros, statistical features etc.

Up to now 19 comparisons have been defined in Simulation News Europe, the series will be continued. Furthermore, a special comparison of parallel simulation techniques has been defined.

2.2. Comparison C2 – Flexible Assembly System

The following example of a flexible assembly system has been chosen because it checks two important features of discrete event simulation tools:

- The possibility to define and combine sub-models.
- The method to describe complex control strategies.

The model consists of a number of almost identical sub-models of the following structure (figure 1):

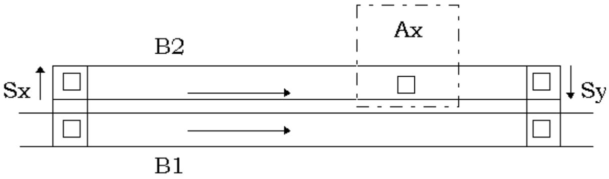


Figure 1: Submodel of assembly station

Two parallel conveyor belts, B1 and B2, are linked together at both ends. An assembly station Ax is placed at B2. Pallets are coming in on belt B1. If they are to be processed in Ax they are shifted in Sx to B2 and possibly enter a queue in front of Ax. If there is no more empty buffer space on B2 or the pallet is not to be processed in Ax it continues its way along B1. Parts that have been processed in Ax are shifted back to B1 in Sy, having priority over those coming from the left on B1.

The total system now consists of 8 of these subsystems, varying in length, operation and operation time (see figure 2). Between two subsequent subsystems there is a space of 0.4 m, whereas pallets from the third subsystem A2 can be shifted directly to A3, and from A6 directly to A1. The shifting parts, however, cannot function as buffers, i.e. a pallet can only enter an Sx if it can leave it immediately.

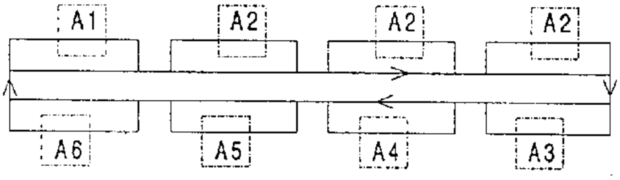


Figure 2: Total System

There are three identical stations A2 in the system, because the operation in A2 takes much longer than the other operations.

Unprocessed parts are put on pallets in A1. They can either be processed in A2 first, and then in A3, A4, A5, or in A3, A4, A5 first, and then in A2. The sequence of operations among A3, A4, and A5 is arbitrary. Station A6 is a substitute for any of the stations A3, A4, A5, i.e. whenever one of these stations is down, or the buffer in front of it is free, the corresponding operation can be executed in A6. Finished parts are unloaded in A1, unfinished parts enter another circle.

3. PETRISIMM TOOLBOX

The so called MATLAB PetriSimM toolbox is based on a basic toolbox (Mušič, Zupančič, and Matko 2003) which deals with analysis, supervisory control synthesis, and non-timed simulation. This basic toolbox is programmed in MATLAB version 5.3 and is therefore adapted to MATLAB version 7.2 (R2006a) to form the MATLAB PetriSimM toolbox. The toolbox is embedded in the MATLAB environment and its usage requires version 7.0 or higher. Furthermore the toolbox is extended with the capability of Timed Petri Nets and timed simulation using the holding durations principle (Löscher, Breitenecker, and Mušič 2005; Löscher, Breitenecker, Mušič, and Gradišar 2005; Löscher, Gradišar, Breitenecker, and Mušič 2006). In another step Coloured Petri Nets are developed for the use in the MATLAB PetriSimM toolbox. The enabling duration principle is added as a second approach of implementing time into Petri Nets. A new way of defining firing sequences is found to be able to model scheduling problems being independent of the occurrence of any conflicts (Löscher and Breitenecker 2006; Mušič, Löscher, and Gradišar 2006). Finally the toolbox is extended with the optimisation of scheduling problems containing heuristic algorithms like Simulated Annealing, Threshold Accepting and Genetic Algorithms (Löscher, Mušič, and Breitenecker 2007). In case of stochastic processes a sequential paired t-test and variance reduction techniques are used and implemented to solve the stochastic optimisation for sequencing and scheduling problems. To sum up, the sophisticated MATLAB PetriSimM toolbox offers the capabilities of analysis, modelling and simulation of Petri Nets. Furthermore it is possible to optimise scheduling problems based on Timed, Coloured, and Stochastic Petri Nets. The open source MATLAB PetriSimM toolbox can be used for education in a graduate level and for modeling and simulating real life processes of discrete event systems in equal measure.

3.1. GUI

Figure 3 shows a screenshot of the graphical user interface (GUI) of the PetriSimM toolbox. The GUI is divided into a menu bar, a button bar and an axes area. In the menu bar different modes can be chosen. The user can switch between analysis, non-timed simulation, timed simulation based on the holding durations principle, and timed simulation based on the enabling durations principle. Furthermore models can be saved, loaded, exported and printed. For each type of simulation several parameters can be set. Another important part of the menu bar is the options menu where the priority wizard and Gantt chart wizard can be started. Next to the options menu the optimisation menu is placed. There several parameters and modes for the optimisation can be changed and selected. The button bar contains several buttons for building, changing, zooming, simulating and analysing the Petri Net models. In the axes area the Petri Net models can be created, simulated and the so called token game can be shown. Figure 3 also contains the Petri Net model of a production cell which is later used for the optimisation.

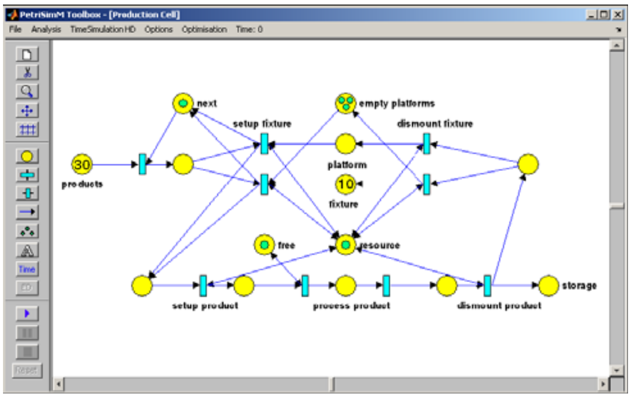


Figure 3: Graphical User Interface

3.2. Simulation

The simulation is a main part of the PetriSimM toolbox. It is separated into non-timed simulation, timed simulation using holding durations and timed simulation using enabling durations. For all three simulation modes an animation of the token game can be visualised. But this feature is only used for educational purposes because through the animation the simulation speed is highly increased. It is possible to change to the animation speed in the parameter section and for the optimisation the animation is deactivated. Another interesting parameter for non-timed simulation is the firing probability. This parameter controls the firing of enabled transitions. This means that it is randomly decided if an enabled transition can fire or not depending on the defined firing probability. For timed simulation time delays can be assigned to the transitions which can be deterministic or stochastic. This means that any probability distribution function can be defined to each transition. For this purpose, any MATLAB m-file can be written resulting to a single positive value, or existing MATLAB probability distribution functions, which can be used to model stochastic time delays. Only the positive part of the used function is taken. If the result of the used stochastic function is negative the time delay is set to zero and a warning is displayed.

3.3. Priority Wizard

Petri nets offer the possibility to handle conditions on the highest level. This is a big advantage compared to the event-oriented tools for modelling and simulation of discrete systems. If conflicts or resource sharing problems occur, a strategy for solving simultaneous events should be available in these tools. Here, Petri nets realize these problems through the basic properties of their structure.

In the PetriSimM toolbox, a Priority wizard allows selecting groups of transitions to define a priority or a sequence for firing. These definitions are the basis for the resolution of conflicts. While in previous versions of the toolbox the firing sequences are dependent on the occurrence of a conflict, the sequence strategy in the current version is independent on conflicts. If a firing sequence is defined for a group of transitions no conflict of the involved transitions is required any more. Only the transition which is defined by the current value of the sequence vector will fire if it is enabled. All other transitions of the group are disabled until this specific transition has fired. This new strategy can lead to a new kind of deadlock if only the disabled transitions are enabled without consideration of the defined firing rules. This new approach offers the possibility to model queuing, sequencing or scheduling problems being independent of the appearance of any conflicts.

3.4. Data Handling

The allocation of the places over time is needed to show and to analyse results of the simulation. This means that the number of tokens has to be stored for each place and for each colour in each iteration step. The number of iterations depends on the specific problem. Therefore this value is not fixed and in general it can be very large. On this account very big matrices are created during the simulation. Dynamic memory allocation could be used to solve this problem. But the access to big matrices causes speed problems corresponding to the size of the used matrices. If the matrices and the stored information are increased too much memory problems can also occur. In the PetriSimM toolbox these problems are solved by the use of external binary files. The built-in MATLAB command fwrite is used to store the data during the simulation. At the end of the simulation the fread command is used to store the matrices in MATLAB as application data of the GUI. If the memory of the program is exceeded the binary files are transformed to text files in order that they can be used for post-processing using other tools or programming languages. The big advantage of this implementation is the fact that the data storing takes the same time in each iteration of the simulation algorithm. The used built-in functions are very fast and therefore only a marginal increase of computation time can be determined for the simulation function.

4. IMPLEMENTATION

A flexible assembly system described in the ARGESIM Comparison 2 (ARGE Simulation News 2008) is an atypical application for Petri Nets. The aim of this simulation and implementation was to prove that even such a process can be simulated by a discrete system based on Petri Nets, although the programming and processing efforts are considerably high. The C2 comparison "Flexible Assembly System" consists of two times two conveyors, eight assembly stations and two shifting parts.

4.1. Use of sub-models

The flexible assembly system is modelled with Petri nets using a discretization approach. The movements of the conveyor belts and the processing steps of the pallets are designed using three places units. A resource place ("free") signalizes whether entrance into the unit is possible or not (figure 4). During the given processing time the tokens pass from the entrance place to the exit place. This is modelled using the holding duration principle.

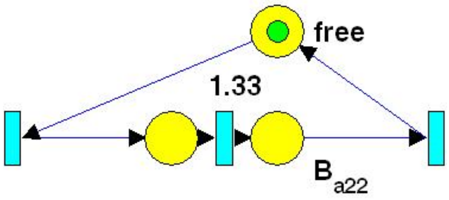


Figure 4: Conveyor unit

Figure 5 shows the corresponding processing unit. For a more concise data recording, the processing unit additionally disposes of a place ("processed") indicating when the pallet is assembled.

An assembly station (figure 6) consists of successive conveyor units and one processing unit, both mentioned above. The PetriSimM toolbox offers additionally the possibility to replicate existing structures and sub-nets. This feature simplifies the building and the creation of each assembly station and the whole flexible assembly system.

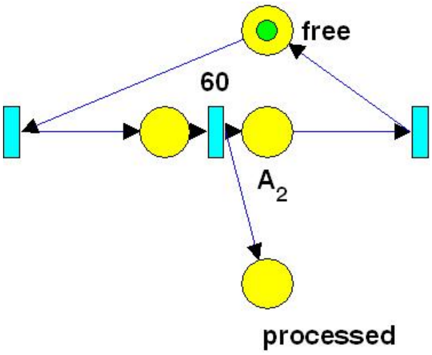


Figure 5: Processing unit

After building the basic structure of the net the connection logic (shifting and switching stations, etc.) of the sub-nets have to be established. Furthermore, a customizing of the different types of assembly stations has to be done.

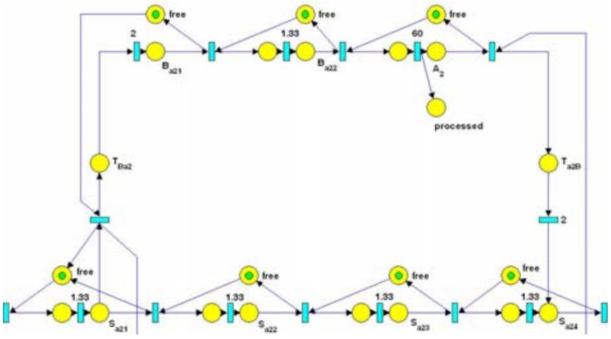


Figure 6: Assembly station

In order to enable simulations with a number of pallets exceeding the band capacity (40), an additional tool had to be introduced: this control unit permits loading of new pallets from the assembly stations to the main conveyor only in case of free capacity and thus avoids system deadlocks.

4.2. Prioritisation

The priorities at the switches in front of each assembly station have been assigned as follows:

1. Leaving the assembly station
2. Entering the assembly station
3. Forwarding along the conveyor

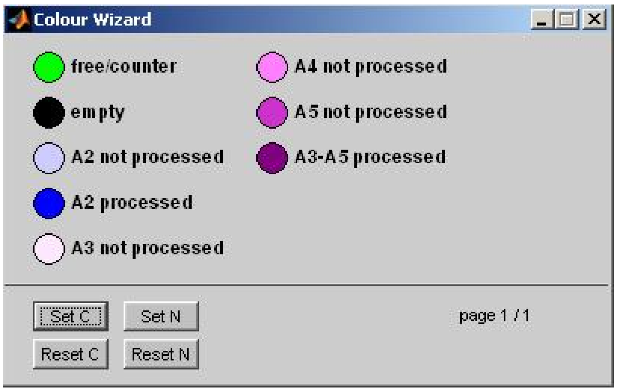
At the end of the conveyor, the shifting part obeys similar rules:

1. Shift directly from station to station
2. Shift from station to conveyor
3. Shift from conveyor to station
4. Shift from conveyor to conveyor

All these prioritisations can be represented as conflicts of the Petri Net. Therefore, priorities are assigned to selected transitions to solve these control strategies by the use of the basic properties of Petri Nets.

4.3. Use of colours

Figure 7 shows a screenshot of the Colour Wizard. The different colours represent the different types of tokens. In the parameter section the number of desired distinguishable tokens can be defined. If more than one token are chosen the so called Colour Wizard can be shown. The Colour Wizard can be used to change the look and the names of the different token colours. The colours are coded as RGB values and can be easily modified. It is also possible to reset all settings to the default values and names. The colours of the used different tokens are interesting for the animation of the simulation. The so called token game is mainly used for educational purposes.



🟢 free/counter
⚫ empty
○ A2 not processed
🔵 A2 processed
○ A3 not processed
● A4 not processed
● A5 not processed
🟣 A3-A5 processed

Figure 7: Colour Wizard

In most discrete simulators and discrete simulation languages the pallets could be modelled as entities. These entities could store information in several different attributes. Petri Nets offer entity simulation in a very rudimental way. Tokens could be used to represent entities of a system but there are not many possibilities to store information. A big disadvantage is the use of the tokens during the firing of transitions. Basically tokens are deleted if they are not needed any more and they are created if they are needed again. A track of information would be very complex and contradict to the definition of place/transition Petri nets.

In this case the colours of the tokens are used to model entities of the system. The pallets are modelled as combination of multiple coloured tokens, the colours making statements about the state of procession of the pallet. One pallet is represented by two coloured tokens. The different colours define the current processing status of the pallet (figure 7).

The Transition Wizard (figure 8) is used to modify and define the transitions in case of Coloured Petri Nets. The graphical description of Petri Nets can be simplified if many identical sub nets are used. Therefore the transitions of the sub nets are merged together and one transition of the Petri Net can represent and contain many other transitions. The Transition Wizard consists of several buttons corresponding to the following functionalities: transitions can be deleted, new transitions can be added, the names of the transitions can be modified and the weights of the corresponding arcs can be defined. Figure 8 shows a Transition Wizard containing the transitions which controls one movement on the conveyor belt. For each transition different weights can be defined for the input and output arcs. These weights control and determine the way of each pallet through the system depending on its current status (colour and numbers of tokens).

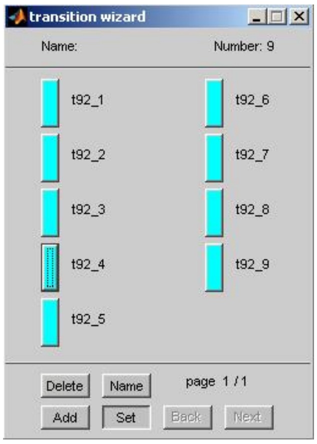


Figure 8: Transition Wizard

5. RESULTS

The complexity of the model (227 places, 157 transitions, 8 colours) results in large scaled input and output matrices (1816 x 929). Figure 9 shows a screenshot of the complete model.

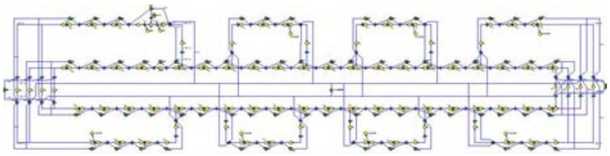


Figure 9: Model

This complexity increase both processing time and quantity of data. Each single simulation took between one and three days on a home computer. The allocation of all places are stored in external log files because the internal memory of MATLAB could not handle this big amount of data and information Furthermore, up to 29 GB of data were produced during one simulation session, in total 110 GB. The processing was therefore delegated to the phoenix-cluster of the University of Technology Vienna.

The evaluation of the ARGESIM comparisons is separated into several tasks. The following sections describe the tasks defined for comparison 2. Based on these results and tasks different simulators and simulation languages can be compared.

5.1. Control Strategy/Statistical Evaluation.

Due to systematic restrictions of the method and the model it is not possible to follow one single pallet throughout the process. The Petri net approach offers entities formed by a combination of coloured tokens but it is not possible to evaluate any statistical data during the simulation. Therefore the information of the allocation of certain places is stored over time in external log files and post-procession of this node data becomes necessary. Figure 10 shows a screenshot of the start and end point of the system. At the beginning and at the end tokens are created in the “in” and “out” places for each pallet, respectively. After the simulation it is not possible any more to get the throughput time for one single pallet but for statistical purposes the average throughput time can be easily calculated. This statistical evaluation was done by external post-processing of the log-files based on small external C programs. Following the recommendation, the data used was from the 120th to the 600th minute of simulation.

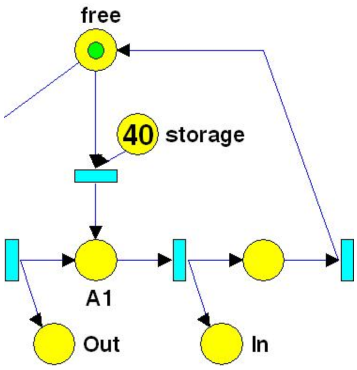


Figure 10: Start and end point

5.2. Simulation Results - Throughput.

The simulation time was eight hours. Table 1 show an extract of the throughput times and the graph of figure 11 aims to provide an overview of the system's behaviour.

Table 1: System behaviour: through put time (extract)

Number of pallets	Total throughput	Average throughput time [s]
5	480	292.50
10	960	292.50
13	1333	273.29
14	**1440**	**272.50**
15	1440	292.50
20	1444	391.92
40	1441	792.04
60	1433	1198.66

Larger calculations than with 40 pallets are purely theoretic, since they exceed the system's capacity. This was made possible by the deadlock control unit.

5.3. Simulation Results and Optimisation.

As shown in the following graph (figure 12), the utilization of stations A2–A6 reaches its possible maximum (100% utilisation of „bottleneck“ stations A2-1–A2-3) starting at 14 pallets. In this configuration, the highest possible throughput (1440 pallets; at some points slightly higher numbers may have encountered due to the delayed measurement) can be obtained at the lowest throughput time (272.50 seconds).

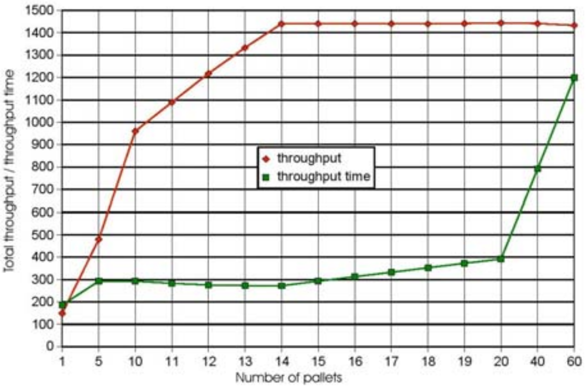


Figure 11: System behaviour – total throughput and throughput time

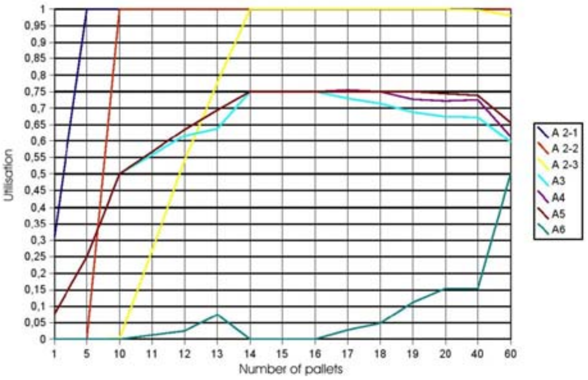


Figure 12: System behaviour: assembly stations' utilisation

6. CONCLUSIONS

The flexible assembly system can be modelled using the advantages and properties of Petri nets and the MATLAB PetriSimM toolbox. Although, the entity approach can not fully represented it is possible to fulfill all requirements of comparison 2 of the ARGESIM series.

REFERENCES

Petri C.A., 1962. *Kommunikation mit Automaten.* Dissertation, Universität Bonn.

Bowden F.D.J., 2000. A brief survey and synthesis of the roles of time in petri nets. *Mathematical and Computer Modelling* 31:55-68.

Zuberek W.M., 1991. Timed Petri Nets. Definitions, Properties and Applications. *Microelectronics and Reliability* 31(4):627-644.

Bause, F., Kritzinger P., 2002. *Stochastic Petri Nets - An Introduction to the Theory (2nd edition).* Germany: Vieweg Verlag.

Zuberek W.M., 1999. Timed Petri Nets in Modelling and Analysis of Simple Schedules for Manufacturing Cells. *Computers and Mathematics with Applications* 37:191-206.

Breitenecker F., 2008. *ARGESIM Benchmarks*. Available from: http://seth.asc.tuwien.ac.at/ argesim/index.php?id=23 [accessed 12 July 2008]

Mušič G., Zupančič B., Matko, D., 2003. Petri Net Based Modelling and Supervisory Control Design in Matlab. *Proceedings of EUROCON Conference*, pp. 362-366. September 22-24, Ljubljana (Slovenia).

Löscher T., Breitenecker F., Mušič G., 2005. Petri Net Modelling and Simulation in Matlab - A Petri Net Toolbox. *Simulation News Europe* 43:20-21.

Löscher T., Breitenecker F., Mušič G., Gradišar D., 2005. A Matlab-based tool for timed Petri nets. *Proceedings of ERK Conference*, pp. 273-276. September 26-28, Portorož (Slovenia).

Löscher T., Gradišar D., Breitenecker F., Mušič G., 2006. Timed Petri Net Simulation in Matlab: A Production Cell Case Study. *Proceedings of MATHMOD Conf*erence, Proceedings on CD. February 7-10, Vienna (Austria).

Löscher T., Breitenecker F., 2006. Petri Net Modelling and Simulation of Production Processes with PetriSimM, a MATLAB-based Toolbox. *Proceedings of 12. ASIM - Fachtagung Simulation in Produktion und Logistik*, pp. 313-319. September 26-27, Kassel (Germany).

Mušič G., Löscher T, Gradišar D., 2006. An Open Petri Net Modelling and Analysis Environment in Matlab. *Proceedings of IMM Conference*, pp. 123-128. October 4-6, Barcelona (Spain).

Löscher T., Mušič G., Breitenecker F., 2007. Optimisation of Scheduling Problems based on Timed Petri Nets. *Proceedings of EUROSIM Conference*, Proceedings on CD. September 9-13, Ljubljana (Slovenia).

AUTHORS BIOGRAPHY

Thomas Löscher studied Technical Mathematics at the University of Technology of Vienna. During his diploma study he specialised in modelling, simulation and optimisation of discrete event systems. In his PhD study he continued his research in the field of simulation-based optimisation. He studied one semester abroad based on cooperation with the faculty of electrical engineering of the university Ljubljana, Slovenia. There he found the basis for his PhD thesis where he optimised scheduling problems based on Timed Petri nets. During his PhD study he also worked for the ARC Seibersdorf research company in the field of discrete event simulation. Currently he is employed as research associate at the Profactor GmbH in the field of simulation-based design and optimisation.

NEW APPROACHES OF COMBINED MODELLING AND SIMULATION IN HEALTH CARE SYSTEMS AND MANAGEMENT

Nikolas Popper[(a)], Michael Gyimesi[(b)], Günther Zauner[(a)] Felix Breitenecker[(c)]

[(a)]"die Drahtwarenhandlung" Simulation Services, Neustiftgasse 57-59, 1070 Vienna, Austria
[(b)]Hauptverband der österreichischen Sozialversicherungsträger (Main Association of Austrian Social Security Institutions), Kundmanngasse 21, 1030 Vienna, Austria
[(c)]Vienna University of Technology, Institute for Analysis and Scientific Computing,
1040 Vienna, Wiedner Hauptstraße 8-10 , Austria

[(a)] niki.popper@drahtwarenhandlung.at, [(b)] Michael.Gyimesi@hvb.sozvers.at, [(c)] Felix.Breitenecker@tuwien.ac.at

ABSTRACT

Modelling and Simulation in Health Economy has two main challenges to cope with. First modelling aspects are widely scattered, dealing with problems in economy, epidemics, medical aspects and more. Second the identification of models is difficult as data sets are in some ways "hided" in different areas (clinical studies, statistics, economic studies, ..), quality and type of given data sets are various and relationships are complicate to identify. This contribution shows the achievements of the cooperation with one of the most important institution in the Austrian health care system, which was started to optimize the solutions of problems mentioned above and to integrate new modelling approaches for analysis of data. An outline of the different aspects like implementation of big data sets in modelling of diabetes mellitus, comparisons of different modelling approaches, combining such approaches to get more effective models and implementing models based on clinical problems should be given.

Keywords: combined simulation, data models, health care modeling, socioeconomics, complex data

1. INTRODUCTION

1.1. Hybrid Simulation Cooperation

Using combined and hybrid simulation in different areas the working group „Mathematical Modelling and Simulation" at the Vienna University of Technology has gained a wide knowledge about solving problems of combining complex data systems and difficult structures of systems to be modelled. Since 2005 the working group works in cooperation with the HVB (Hauptverband der österreichischen Sozialversicherungsträger, Abt. Gesundheitsökonomie - Main Association of Austrian Social Security Institutions, Dept. Health Economy) to implement this knowledge in the area of Health Care Modelling.

1.2. Applications & Structure

A main goal is to help the HVB to supply information and knowledge to implement an information platform. This platform should be able to show differences between the current situation and target values and to allow an analysis of complex situations and the identification of problematic situations in different areas of the health care system. Potential fields of activity for actions in the area of the Social Security Institutions or other areas should be identified, mentioned or at least be described. For these tasks appropriate methods and strategies should be provided. Another main goal is to transfer the results to different areas (process owners and decision makers) inside the Social Security Institutions (Figure 1).

Various methods of discrete and continuous modelling and simulation are used and a huge amount of structured or unstructured data sources have to be analysed. The combination of methods like system dynamics, cellular automatons, agent based modelling, differential equations and other modelling techniques have to be compared and sometimes apply within one subsystem. Various Data models have to be mapped, basing on a very complex system of the HVB and the different social security institutions.

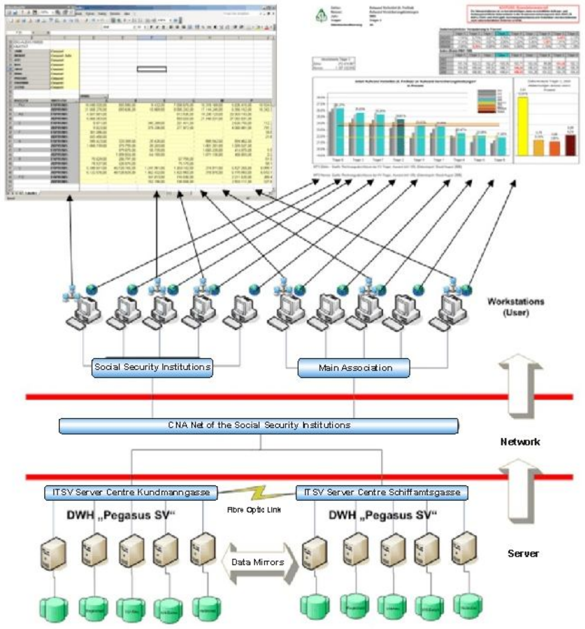

Workstations (User)
Social Security Institutions
Main Association
CNA Net of the Social Security Institutions
Network
ITSV Server Centre Kundmanngasse
Fibre Optic Link
ITSV Server Centre Schiffamtsgasse
DWH „Pegasus SV"
DWH „Pegasus SV"
Data Mirrors
Server

Fig. 1: Structure and Technical Foundation of the Provided Data Sources

A main goal of the project is to improve the state of the art of modelling approaches in health economy to improve the opportunities in the decision processes. For this reason new data models, new combinations of modelling approaches and an improved possibility of parameter identification have to be provided.

1.3. Subsystems and Model Approaches

As application examples different approaches will be described. First there is the implementation of a System Dynamics model for the incidence of type-2 diabetes in Austria based on a model developed by A. Jones, J. Homer et al. for the United States of America. In this case a huge amount of given data sets had to be imported and integrated into the model. The comparison of classical differential equation methods and cellular automatons for analysis of epidemic models led to results for advantages and disadvantages of the different models. These and other results influenced a combined approach (Cellular Automatons and Agent Based Simulation) for modelling and simulating inhomogenous communities to analyse epidemic influenza scenarios. Finally the implementation of e-information systems to provide information for physicians to cope with PSA measuring.

2. MODELS & IMPLEMENTATIONS

2.1. Using System Dynamics in modelling Diabetes mellitus

Diabetes mellitus (DM) and its complications are one of the most challenging topics in public health care. We adopted a System Dynamic (SD) model, commissioned by the Center for Disease Control and Prevention (CDC) in the USA, which has been successfully applied to reproduce the historical available data of the last two decades. The structure of the model arises not only from the progression of DM as a chronic disease but also from the available data. We adopt the model to the Austrian data set and enhance it to include a distinction by sex since different policies may become necessary.

2.1.1. Decision of Modelling Technique

There were different aspects for the choice of the modeling technique:

1.) Different players in health economy recognize the threat and agree that measures on an population-wide, system-wide level have to be taken to reduce chronic diseases and their consequences. But most programs use conventional analytical methods by which each aspect of a complicated disease control strategy is addressed and evaluated separately. The advantage of SD here is that one gets a global picture where all influencing factors are incorporated and act together.

2.) As chronic diseases involve long time scales there are long delays between causes and health consequences making short term analysis methods unsuitable. Three prevention levels, of which each can require dozens of years of treatment, are distinguished: primary prevention to avoid the onset of an affliction, secondary prevention to avoid chronic development and harmful consequences and tertiary prevention to avoid the loss of functions.

3.) For every prevention level many different policies are available. Primary prevention includes behavioral and socioeconomic measures like improving lifestyle, working and living conditions, information, education and many more. Secondary prevention focuses on precaution and early detection. And finally elements of the tertiary prevention are accessibility to the medical treatment, improvement of compliance and empowerment. SD now gives the opportunity to test different approaches and policies simultaneously and observe the respective outcome.

2.1.2. Modelling Approach

In Figure 2 the population stocks and flows in the model are shown. Seven different population stocks are arranged in four groups. The first group consists only of one stock: the healthy adults who have a normal blood-glucose level. The other groups each consist of two levels, the diagnoses and the undiagnosed ones. The second group is the population with pre-diabetes. These are people with an increased blood-glucose level but not yet having developed full diabetes, which constitute the third group. In the last group are people who not only have diabetes but are also stricken by consequent diseases.

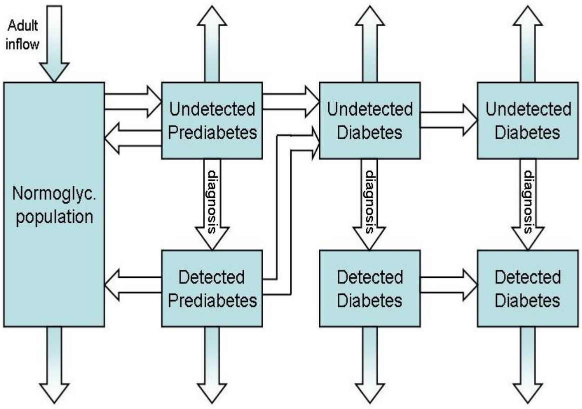


Fig. 2: The main stocks and flows of the model

There is only one inflow of healthy adults into the first level, while people may die out of every level. This inflow is given as a time series input by statistical predictions. The different death rates are affected by the fraction of obese people of every stock, which is calculated in our model, as well as by the fraction of elderly people, which is again given as a time series. The basic assumption is that the relative rates of people with a risk factor compared to people without it remains constant in the respective group. Written explicitly

$$\begin{gathered}\frac{\mathbf{P}(\text{death} \mid \text{elderly})}{\mathbf{P}(\text{death} \mid \neg\text{elderly})} = const. \\ \frac{\mathbf{P}(\text{death} \mid \text{obese})}{\mathbf{P}(\text{death} \mid \neg\text{obese})} = const.\end{gathered} \tag{1}$$

holds true for every group, where **P**(a|b) denotes the conditional probability of factor a under condition b. If DM is already detected than also the control of the disease, the "disease management", is influencing the death rates. With suitable initial values the dynamic death rates can then be calculated.

The flows between the different stocks are characterized by the following assumptions: While people with pre-diabetes can still recover, there is no way to cure DM after its onset. DM is a chronic disease after all and once complications occur the damage is dealt and cannot be undone. The onset of pre-diabetes and DM occur unobserved, while complications can also arise even if under medical supervision. All transition rates are affected by the elderly and the obese fractions of the respective populations. The progression rates (the horizontal untitled ones) of the detected populations can be influenced by the clinical management, like prevention measures and compliance. The detection rates (the vertical ones) are more difficult to describe: they are first order exponentially delayed functions of the progression rates as well as the testing frequency and the sensitivity of the tests. Time dependent input data enter in several places of DM detection and control incorporating different possible health policies.

2.1.3. Data Integration

In the original model there are over 134 different input parameters and not all of them can be measured directly. It is therefore necessary to estimate some of the unmeasured input parameters so that the output reproduces available historical data. This is the reason why we start the simulation in 1980 and continue it till 2050.

One major difficulty encountered when modeling diseases in general is the estimated number of unreported cases. Our findings are in fairly good agreement to the WHO estimates of a current DM prevalence of 5 to 7 percent. The exact number of cases is not to be taken intimately, but this isn't our goal anyway. In the application we want to compare different policies of health care management against each other.

For the analysis of the model we use data for Austria, since the quality of the data is very good and many input parameters are available, especially with respect to the distinction by sex. This distinction is made by running the model twice with different input parameters and then adding up the respective results.

The most important influencing factors are: age, clinical management and obesity. The age enters through the fraction of elderly people. The adult population, that is age 20 and above is given by a time series. The fraction of elderly people is calculated as the fraction of people age 65 and above compared to the total populatio. The calculation of the values for each year is done by a spline interpolation of order 3 of the available data. The results change less than one percent if linear interpolation is used.

2.1.4. Testing of Policies

A main part of the model was to develop a system for testing different policies in a kind of qualitative way. As an example for such a possible policy testing we calculated the results of the same test run with the difference that people spend an additional 200 kcal per day.

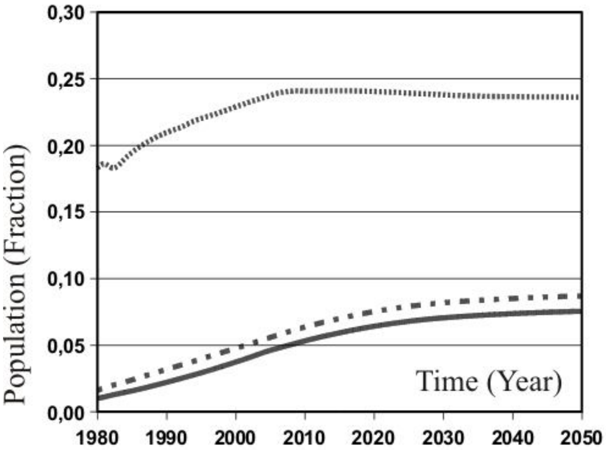


Fig. 3: Pre-diabetes (dotted), diabetes (slash-dotted) and detected diabetes (solid) fractions of the adult population in Austria

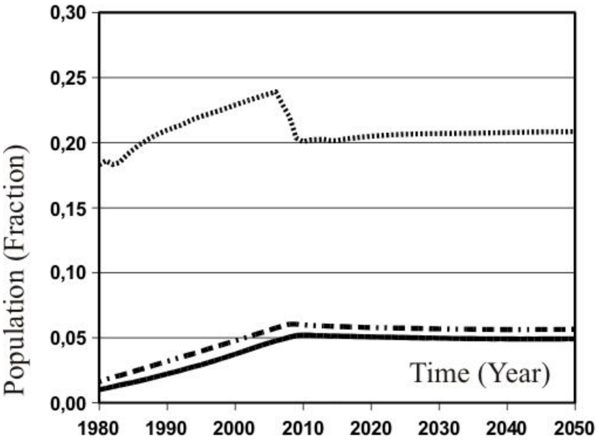


Fig. 4: Same as above - spending additional 200 kcal per day after '05

In Figure 4 we see the same data as in Figure 3 with the difference that people are going for a walk at moderate speed for approximately an hour per day from 2005 onwards.

We see that the growth of the DM percentage stops almost immediately and the pre-diabetes fraction exhibits a sharp drop. The total numbers however are still increasing and only due to a faster increase in the adult population we get a slow decline in the DM rate. This example is somewhat academically since it may not be incorporated in reality. However it shows very nicely that a step-like change in the entry produces a delayed output. While the fraction of the pre-diabetes population is already falling, the fraction of the people with diabetes continues growing for a few more years.

A main goal of the project was to simulate different regions of health care to analyze the west-east gradient of life expectancy and life style in Austria. Together with the HVB and other public decision makers responsible for health care various policies may be tested. Especially interesting is whether different policies for men and women are useful. Ongoing studies to examine the disease management from prevention and early detection over lifestyle adjustment to compliance are suited to validate the predictions made by the SD model.

2.2. Comparison of Different Approaches for a SIR-Epidemic

2.2.1. Problem

Models for the spread of epidemics are nearly as old as the mathematical theory of differential equations. The classical methods applied for modeling such epidemics used to be ODE -Systems but unfortunately these systems are limited in some respect. They become particularly complicated or insufficient when spatial components of disease propagation should be taken into account. This corresponds to the fact that the basic assumption of the classical Kermack – Mc Kendrick (in 1926) model, where the numbers of involved individuals (S, I, R) are described by the following system of ordinary differential equations (ODE).

$$\begin{aligned} \frac{\partial S(t)}{\partial t} &= -\alpha S(t)I(t) \\ \frac{\partial I(t)}{\partial t} &= \alpha S(t)I(t) - \beta I(t) \\ \frac{\partial R(t)}{\partial t} &= \beta I(t) \end{aligned} \tag{2}$$

is a homogeneous population. Spatial inhomogeneities become especially relevant when vaccination strategies or partitions of the population are observed. In this approach different types of introducing spatial patterns into these dynamics were compared. The applied techniques cover lattice gas cellular automata (LGCA), stochastic cellular automata (SCA) and partial differential equations (PDE).

2.2.2. Different Modelling Techniques

A first step towards to an extended model with better spatial behavior was to define a detailed lattice gas cellular automaton (FHP-LGCA). This approach was validated for the classical structure with the ODE – System. Furthermore vacation strategies where tested and we see that we get better behavior for the whole system.

By stochastic cellular automata (SCA) we refer to ordinary cellular automata (without considering motion of particles) with a stochastically determined neighbourhood. To define a “sociological” neighbourhood in the SCA model, that provides a gradation for the occurrence of interaction between individuals, a decaying likelihood of interaction between cells depending on the distance between them can be used. This approach delivers a radial-symmetric distribution of the contacts for each cell, what principally can be described by an arbitrary probability distribution or a similar function, which we denote likelihood functions.

It is not difficult to show that this approach extends the classical model by a spatial component. The ODEs are again an upper bound concerning the speed of spread and a lower bound for spatial inhomogeneities. If the SCA establishes contact between each two cells (dissolution of the local character and increase of speed), a probability of infection for every individual is obtained.

Finding a connection to the LGCA approach on the contrary is not straight forward. It is possible to find some rules of thumb concerning the weight factors for the SCA, but these rules deliver rather imprecise results and only apply to specific conditions. The difficulty lies in finding a tool that relates motion of particles in the LGCA with distributed contacts in SCA.

Despite this serious difference, we can always find appropriate parameters for the stochastic CA to fit the behavior of the epidemic in the LGCA model. By introducing cyclic motion in the HPP automaton, we minimize the direct interaction area of each individual and thus can easily find a corresponding radius of interaction and the appropriate number of contacts per time step for the stochastic CA. We see, that even without this modification – if the motion in the FHP automaton is determined by standard FHP-I transition rules for example – we can find parameters, which deliver the same quantitative and qualitative behavior. The same is true for the classical SIR model, which requires an infinite radius of interaction and an infinite number of contacts per time step.

Another completely different approach towards an extended SIR model can be by partial differential equations. A modified heat-conduction equation and a suitable discrete solution method was defined with a second order Taylor - polynomial for model simplification. This two dimensional model also fits the time behaviour of the spread of an epidemic in a better way than the classical Kermack – Mc Kendrick equations. We compare the diffusion of the infections from the PDE model with the spread of epidemics in CA models. The advanced CA model is simpler to adapt and fits the real behaviour of an epidemic in an adequate way.

2.2.3. Results

The key output value was the number of currently infected individuals but also a visualisation of the density of infected individuals on the domain permits conclusions on the model behaviours. The input parameters involved the size of the domain (grid size), the number of performed time steps, the disease stage transition parameters, the number of runs used for generating an averaged result (Monte Carlo method) and the parameter *t* of the diffusion distribution that was used in the PDE approach. Results are shown in Fig. 5.

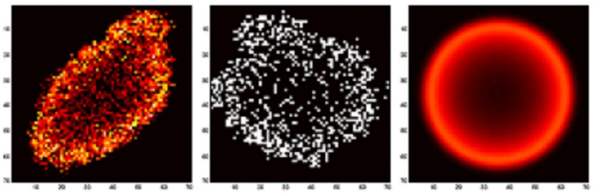


Fig. 5. A lattice representation of the FHP LGCA approach (left), the SCA approach (centre) and the PDE approach (right). Light pixels mark high densities of infected particles and dark pixels mark low densities. This figure does not show specific results but is intended to demonstrate general lattice representations.

An optimal model identification would persist, if the differences are minimal for arbitrary parameter settings. In our case there exist two (heuristic) parameter regions, which deliver good respectively bad correspondence in the model behaviours.

Even though all these methods involve distinct types of spatial interaction, it can be shown, that consistent qualitative and quantitative model behaviour can be obtained by means of parameter adaptions and slight technical modifications. These modifications are motivated by stochastic analysis of distributed interaction (PDE, SCA) and diffusion dynamics (LGCA) as well as prevailing physical analogies. The law of large numbers permits to approximate stochastic contacts by distributed interaction. Diffusion of particles can be approximated through empiric adjustment of a Gaussian diffusion distribution.

2.3. Combined Modelling for Analysis of Influenza Epidemics

2.3.1. Modelling Influenza

In the course of the project to analyse influenza epidemics a hybrid mathematical model was established. The classic methods applied for modelling such epidemics used to be ODE-Systems but unfortunately these systems are limited in some respect. They become particularly complicated and complex beyond limit when observing heterogeneous populations and spatial components. Thus the potential of alternative approaches – namely cellular automata (CA) and agent based systems (AB) – is analysed in the beginning of this work.

Analysis of these methods was split into two major parts. The first one being the theoretical one in which the methods were compared in order to locate their respective strengths and weaknesses. The second part being the practical analysis including behaviour of the implementations.

2.3.2. Combining CA and AB Simulation

In the model an average persons day is divided in three major parts being working time or time at school, leisure time and social life and as a third part time at home respectively sleeping. At the work place, child care facility or school a person is going to meet the same people every day. During leisure time a person usually visits friends, doctors, goes shopping and so on and usually stays within a defined surrounding. The people one meets during this time are often the same. Finally being at home it is assumed that only the family is together. There are no long-distance contacts included. We also do not consider all contacts during the leisure time but simply replace them by a “neighbourhood” of random people.

An Agent based approach to control the whole system and cellular automata to model the subsystems (schools, working places, neighbourhoods, etc.). Such a structure would allow our agents to switch easily from one sub-system to another, without any time gaps within the model. At the beginning of the model the population is randomly initialized with the parameters derived from demographic data. This means that every agent does have a unique ID, a certain health state, an age, a work place (or child care facility respectively school) depending on its age, an assigned household and neighbourhood. Every day all agents move to their work places (respectively schools or child care facilities) and spend the working time there. Excluded from this procedure are senior citizens which are assumed to stay at home during this time. The simulation of the working time is done by cellular automata: every workplace is simulated in a separate automaton. This is convenient to implement and offers great potential for parallelization. This is becoming specially interesting in the near future with increasing numbers of cores on CPUs.

After work the agents proceed to the neighbourhoods which are again simulated separately by cellular automata, thus parallelization is applicable here as well. It would be possible to process several households/neighbourhoods parallel on machines with multiple processors or cores and by this improve the performance of the model. At the end of the day the agents return into their households. Here infection is simulated by simple probabilities since contact between all members of the household can be taken as given. In single households infection is of course not possible

2.3.3. Scenarios

As the advantages are described in chapter 2.2. also the main advantage of this implementation is the possibility to experiment with different policies and even more important in this case with given data sets from the “real life”. As an important aspect the model can relative easily be adapted to new findings an results from our cooperation partner. As one quantitative result we found, that the number of infected can be reduced by 45% if every fifth already infected stays at home.

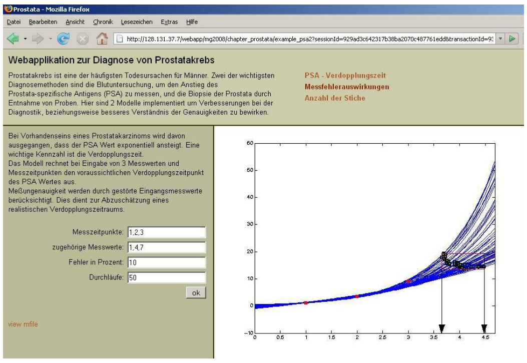
2.4. PSA value changes over time

Additional a project is mentioned that is based on the observation of a quasi – exponential ascent of PSA (prostate specific antigen) – value during affection on prostate cancer.

The input parameters of the model are three measurements of the marker in the blood of the patient at different times. An important appraisal of the characteristics of the illness is the so called doubling time of the PSA value. Furthermore, after ablation of the prostate, it can be tested with this marker, if parts of the afflicted tissue are still in the body of the patient.

As it is nearly impossible for human beings to find the exact exponential fitting curve through three measurement points and to see when the doubling time of the last measurement occurs, computers are used for calculation. For this reason an internet based tool for assistance in analysis of the behaviour was developed for the users. Important additional features of the new tool, comparing with other systems developed in the USA, are the graphical output of the results for a better comprehension also from the patients and the comparison with a linear approximation curve, which is a common effect for elderly patients. Starting with three measurements a nonlinear optimization with the principle of least square error method is implemented.

One of the main tasks and improvements to other methods is to question the reliability of the results. This second task is dealing with the minimal and maximal doubling time depending on the occurrence of measurement errors.



Webapplikation zur Diagnose von Prostatakrebs

Prostatakrebs ist eine der häufigsten Todesursachen für Männer. Zwei der wichtigsten Diagnosemethoden sind die Blutuntersuchung, um den Anstieg des Prostata-spezifische Antigens (PSA) zu messen, und die Biopsie der Prostata durch Entnahme von Proben. Hier sind 2 Modelle implementiert um Verbesserungen bei der Diagnostik, beziehungsweise besseres Verständnis der Genauigkeiten zu bewirken.

Bei Vorhandenseins eines Prostatakarzinoms wird davon ausgegangen, dass der PSA Wert exponentiell ansteigt. Eine wichtige Kennzahl ist die Verdopplungszeit.
Das Modell rechnet bei Eingabe von 3 Messwerten und Messzeitpunkten den voraussichtlichen Verdopplungszeitpunkt des PSA Wertes aus.
Meßungenauigkeit werden durch gestörte Eingangsmesswerte berücksichtigt. Dies dient zur Abzuschätzung eines realistischen Verdopplungszeitraums.

Messzeitpunkte: 1,2,3
zugehörige Messwerte: 1,4,7
Fehler in Prozent: 10
Durchläufe: 50

Fig. 6 Solutions with error afflicted input data, the two arrows in the right section of the figure, point out the minimal and maximal doubling time, as calculated for the exponential growth model assumption

The result, which is calculated through solving the solution routine with different randomly disturbed measurement data streams, is visualized by plotting all exponential interpolation curves and an interval, whereby the left boundary is the minimal doubling time and the right sides is the maximum doubling time depending. This is done to help the user defining the next date for new measurements.

2.5. Modelling and Simulation Results

By now the project showed that the possibilities of parameter identification and reaction times for integrating new data sets could be improved. As the diabetes model showed the possibility of working together the models for epidemic problems or technical solutions were the first proves for solving concrete problems and improving the possibilities of modelling and simulation in this field.

ACKNOWLEDGMENTS

The cooperation which led to the described models is partially funded by the Hauptverband der österreichischen Sozialversicherungsträger, Abt. Gesundheitsökonomie - Main Association of Austrian Social Security Institutions, Dept. Health Economy

REFERENCES

Homer J., et al, 2004, *The CDC diabetes system modeling project*, 22nd International Conference of the System Dynamics Society, Oxford, England

P. Kristöfel, P., 2007, *A System Dynamics Model for the Diabetes Prevalence in Austria*, Diploma Thesis, TU Vienna

Wolf-Gladrow, D.A., 2000, *Lattice-Gas Cellular Automata and Lattice Boltzmann Models*, New York, Springer

Venkatachalam S., Mikler A. R., 2005, *Towards Computational Epidemiology: Using Stochastic Cellular Automata in Modeling Spread of Diseases.* Proceedings of the 4th Annual International Conference on Statistics, Mathematics and Related Fields, January 2005, Honolulu

Nixon R.G., Wener M.H., Smith K.M., Parson R.E., Strobel S.A., Brawer M.K.. 1997, *Biological Variation of Prostate Specific Antigen Levels in Serum: an Evaluation of day-to-day physiological Fluctuations in a well-defined cohort of 24 Patients, The Journal of Urology*, Vol.157, 2183-2190.

AUTHORS BIOGRAPHY

Nikolas Popper has earned a degree in technical mathematics at the Vienna University of Technology. His thesis title is "Simulation of the Respiratory System - Compartment Modelling and Modelling of Perfusion". After research stays abroad in Barcelona, Universitat Politècnica de Catalunya and Moscow, Idaho, University of Idaho, he has experience in industry projects as well as research and development knowledge. Currently he is working in the area of visualization in computer graphics, modeling and simulation of health economy and theory of modeling & simulation. He is co-proprietor of the company “die Drahtwarenhandlung” Simulation Services. The company offers as well technical solutions (defect detection on pictures, modeling & simulation, …) as well as animations and films in the area of Science Journalism. Furthermore he is doing a PhD thesis in the area of alternative and coupled models at the Vienna University of Technology.

A PETRI NET MODEL FOR AN ACTIVE DATABASE SIMULATOR

Joselito Medina-Marín[(a)], Xiaoou Li[(b)], José Ramón Corona-Armenta[(c)], Marco Antonio Montufar-Benítez[(d)], Oscar Montaño-Arango[(e)], Aurora Pérez-Rojas[(f)]

[(a) (c) (d) (e) (f)]Autonomous University of Hidalgo State, Advanced Research in Industrial Engineering Center, Carr. Pachuca-Tulancingo Km 4.5 Col. Carboneras, Pachuca, Hidalgo, México
[(b)]Research and Advanced Studies Center of the National Polytechnic Institute, Av. Instituto Politécnico Nacional No. 2508, Col. Zacatenco, México, D.F., México

[(a)]jmedina@uaeh.edu.mx, [(b)] lixo@cs.cinvestav.mx, [(c)]jrcorarm@yahoo.com, [(d)]montufar@uaeh.edu.mx,
[(e)]oscarma11@hotmail.com, [(f)]auropr@yahoo.com,

ABSTRACT

Active database systems were introduced to extend the database functionality. As well as a repository of data, active database can detect the occurrence of events in a database system and react automatically to event occurrence and execute certain actions either inside or outside the database. This behavior is specified by means of ECA (event-condition-action) rules, i.e., when an event has occurred, if the condition is evaluated to true, then an action is executed. In this paper a simulator for active databases, named ECAPNSim, is described. ECAPNSim uses the definition of ECA rules like a structure of an extended Petri net model, the Conditional Colored Petri Net (CCPN). Conditional Colored Petri Net definition involves the knowledge and execution model, which describe the features that an active database system must have. An example has been developed in order to show the ECAPNSim applicability in a certain study area.

Keywords: Petri nets, active database, ECA rule, simulation.

1. INTRODUCTION

Traditional databases (DB) were developed to store a huge amount of information. In this DB type the information only is accessed by insert, delete, update and query algorithms, which were previously programmed in a Data Manipulation Language (DML) by the DB administrator. The set of all this data manipulation programs is the Database Management System (DBMS). However, the execution of those programs is performed only by the request of either a DB user or the DB administrator.

Nevertheless, there are systems that cannot be implemented by using a traditional DB approach. Such systems are those where is well known that if certain events occur in the DB and if the DB state satisfies certain conditions, then an action or procedure is performed in the DB. Therefore, it is necessary to use an approach where a DB could have the ability to react automatically when an event occur either inside or outside DB environment, after this, it can verify the DB state to evaluate conditions, and if condition is evaluated to true it can execute procedures that modify the DB state. In order to provide of active behavior to traditional DB, Active Databases (ADBs) were introduced. If a human being takes charge to detect the event occurrences, verify conditions, and execute procedures instead an ADB system, then the system may not work well. Thus, it is very important to add enough information to DB about the active behavior and convert a traditional DB into an Active one.

Active behavior of a DB can be defined through a base of active rules, which has the specification of events that will be detected, conditions that will be evaluated, and actions or procedures that will be performed in the DB. The model most widely used is the event-condition-action rule (ECA rule) model, whose general form is as follows (Silberschatz, Korth, and Sudarshan 1999):

on event e_1
if condition c_1
then action a_1

ECA rule model works in the following way: when an event e1 that modifies the current DB state occurs, if condition c1 is evaluated to true against DB state, then either an action a1 is executed inside DB or a message is sent outside DB.

An event e1, which can trigger to an ECA rule, can be of two types: primitive event or composite event (Paton and Diaz 1999). A primitive event is generated by the execution of an operation over the DB information (insert, delete, update, or select), a DB transaction, a clock event (which can be absolute, relative, or periodic), or the occurrence of a DB external event. On the other hand, composite events (disjunction, conjunction, sequence, closure, times, negation, last, simultaneous, and any) are formed by the occurrence of a combination of primitive and/or composite events.

Composite events increase the complexity of a base of active rules because composite events are represented by complex structures, which need to be evaluated when a composite event is raised. In the same way that a composite event increases the complexity of a base of active rules, relationships between ECA rules increase the complexity of a base of active rules.

Furthermore, active rules must be validated before its implementation into a real active database system, in order to know its behavior and to verify the presence of situations that may produce an inconsistent state in the database system.

This verification can be performed through the simulation of the active rules. In this paper an ECA rule simulator is presented, which uses a Petri net model, named Conditional Colored Petri Net (CCPN), to depict ECA rules as a Petri net structure, and with the token game animation the event occurrence and rule triggering are analyzed in order to detect active database problems such as No termination and confluence (Paton and Diaz 1999).

The remainder of the paper is organized as follows. Section 2 discusses the related work about active database systems; section 3 gives a general description of the PN model used to depict ECA rules. Section 4 describes a software development named ECAPNSim, which is used to model, simulate and analyze ECA rule sets. Section 5 shows the applicability of ECAPNSim by developing an example of ECA rule set in a bank enterprise. Finally, Section 6 concludes and gives directions for future work.

2. RELATED WORK

There are several research studies about active databases and the development of ECA rules. Relational systems, such as starburst (Widom 1996), Postgres (Stonebraker and Kemmintz 1991), Ariel (Hanson 1996), SYBASE (McGoveran and Date 1992), INFORMIX (Lacy-Thompson 1990), ORACLE (Hursh 1991), among others, provide an active functionality based on triggers, but they cannot handle composite events at all.

Triggers only supports the composite event disjunction, and structure primitive events that are defined over a table, moreover, in the action part of triggers cannot be executed another trigger.

On the other hand, Object Oriented DB systems (such as HiPAC (Dayal, Blaustein, Buchmann, Chakravarthy, Hsu, Ledin, Mc-Carthy, Rosenthal, Sarin, Cary, Livny, and Jauhari 1998), EXACT (Paton and Diaz 1999), NAOS (Collet and Coupaye 1996), Chimera (Ceri, Fraternali, Paraboschi, and Tanca 1996), Ode (Gehani and Jagadish 1996), Samos (Gatziu and Ditrich 1999)) provide more elements of active systems, like the composite event handling. Nevertheless, because of the different structures and classes used to develop Object Oriented DB systems, there is not a standard model to define ECA rules in these systems.

Few researches have adopted Petri nets as ECA rule specification language (Gatziu and Ditrich 1999), (Li, Medina-Marín, and Chapa 2002) (Schlesinger and Lörincze 1997).

Colored Petri Nets (CPN) are a high-level Petri nets which integrate the strength of Petri nets with the strength of programming languages. Petri nets provide the primitives for the description of the synchronization of concurrent processes, while programming languages provide the primitives for the definition of data types and the manipulation of their data values (Jensen 1994). So it is more suitable for active database than ordinary Petri nets since it can manipulate data values. By using CPN one can not only revealing the interrelation between ECA rules but also capture the operational semantics. For these reasons, CPN is very suitable for modeling and simulation of active rules. (Schlesinger and Lörincze 1997) adopted CPN as rule specification language, and they proposed an Action Rule Flow Petri Net (ARFPN) model, and a workflow management system was illustrated to verify their ARFPN model. However, there exists much redundant PN structure for using ”begin of”, ”end of” events, conditions and actions repeatedly. So, Their CPN model is very large even for a small rule set. Therefore, the complexity of CPN management increases. In SAMOS (Gatziu and Ditrich 1999) a SAMOS Petri Nets (S-PN) was proposed for modeling and detection of composite events. S-PN is also CPN-like where a different perspective for colors was taken. Colors in SPN are token types, and one token type is needed for each kind of primitive event; however, the framework is not Petri-net-based.

3. CONDITIONAL COLORED PETRI NET DEFINITIONS

There are several proposals to support reactive behaviors and mechanisms inside a DBS, which is best known as an ADBS. Nevertheless, these proposals are designed for particular systems, and they cannot be migrated to any other system, moreover, there is not a formal ADBS proposal.

In this paper, a general model to develop ECA rules in an ADBS is proposed, based in PN theory, which can be used as an independent engine in any DBS. An ADBS must offer both a knowledge model and an execution model. Knowledge model specifies the elements of the ECA rule, i.e., the event, condition, and action part. On the other hand, execution model describes the way in that the ECA rule set will be executed.

In knowledge model, each ECA rule element is converted into a CCPN element. The event, which activates the ECA rule, is converted in a CCPN structure that is able to perform the event detection. A Primitive event is depicted by a CCPN place, but if the event rule is composite, then the corresponding CCPN structure is generated. Both types of events finish in a place, which will be used as an input place for a transition.

A CCPN transition holds the next element of an ECA rule, the conditional part. It verifies if there are tokens in its input place and evaluates the conditional part of the ECA rule that is holding. Unlike traditional PN transitions, CCPN transitions have the ability to evaluate boolean expressions.

Finally, the ECA rule element action. When action part is executed in a DBS, it modifies the DB state. This can be viewed as an event that modifies the DB state. Events are represented as CCPN places, thus action part is represented by a place too. The difference between places for events and places for actions is that places for events are input places to transitions, and places for actions are output places from transitions.

CCPN execution model is based in the transition firing rule of PN theory. It provides mechanisms to create tokens with information, or color, about events that are occurring inside the DB. New tokens are placed in the corresponding places for those events. This is the way in that an ECA rule set is processed and both composite and primitive events are detected. By using Colored Petri Nets (CPN) is possible to depict ECA rules, but only those that have primitive events. ECA rules with composite events cannot be represented efficiently with CPN.

Conditional Colored Petri Net (CCPN) (Medina-Marín 2005b) is a Petri net extension, which inherits attributes, and transition firing rule from classical PN (Li, Medina-Marín, and Chapa 2002) (Li and Medina-Marín 2004) (Medina-Marín and Li 2005a). Furthermore, CCPN takes concepts from the CPN, such as data type definition, color (values) assignation to tokens, and data type assignation to places.

In the CPN case, data type assignation is performed for all the places of CPN, on the other hand, in the CCPN case, data type assignation for places is not general, because the CCPN handles a kind of place (virtual place – denoted as dashed circles) with the ability to hold different types of tokens. Tokens can denote different data structures, according to the tables of databases.

In order to evaluate conditional part of ECA rule stored inside a CCPN transition, a function is defined to do this task. Evaluation function analyzes the boolean expression and matches it with the DB state to determine its boolean value.

Some composite events needs to verify a time interval, hence CCPN provides a function that assigns time intervals to a CCPN transition, which will be the responsible to verify if events are occurring inside time interval defined, likewise the evaluation of ECA rule condition is done. These types of transition are named composite transition.

Each event occurs in a point of time, thus, CCPN provides a functions that assigns a time stamp to every token created. Time stamp value is the time instant in which the event has occurred. It is useful to verify if an event occurred inside a time interval or to detect composite events such as sequence and simultaneous.

Finally, every time that an event occurs, a token must be created. CCPN has a function to initialize tokens, in other words, when an event occurs in DB, a new token is created by CCPN and its attributes are initialized to the corresponding event values. The new token is put in the place that represents to detected event.

CCPN is an extension of PN that uses CPN concepts (Jensen 1994). In order to save event information in tokens and to create new tokens with data about the action part of the ECA rule, CCPN uses the concept of "color" taken from CPN. The values stored in tokens are used to evaluate the conditional part of the rule stored in the transition of CCPN. CCPN uses the multi-set concept from CPN, because a CCPN place may have several events at the same time. Unlike CPN, CCPN evaluates conditions inside transitions; meanwhile CPN evaluates conditions in its arcs.

Formally, a CCPN is defined as follows:

Definition 1. A conditional colored Petri net (CCPN) is a 11-tuple

CCPN = {Σ, P, T, A, N, C, Con, Action, D, τ, I}

where

(i) Σ is a finite set of non-empty types, called color sets.

(ii) P is a finite set of places. P is divided into subsets, i.e., $P = P_{prim} \cup P_{comp} \cup P_{virtual} \cup P_{copy}$, where P_{prim} represents primitive events and it is depicted graphically as a single circle. P_{comp} represents composite events negation, sequence, closure, last, history, and simultaneous and it is depicted graphically as a double circle. $P_{virtual}$ represents composite events conjunction, disjunction, and any and it is depicted graphically as a single dashed circle. And, P_{copy}, is a set which is used when two or more rules are triggered by the same event and it is depicted graphically as a double circle where the interior circle is a dashed one.

(iii) T is a finite set of transitions. $T = T_{rule} \cup T_{copy} \cup T_{comp}$., where T_{rule} represents the set of rule type transitions and it is depicted graphically as a rectangle. T_{copy} is the set of copy type transitions and it is depicted graphically as a single bar. T_{comp} is the set of composite type transitions and it is depicted graphically as a double bar.

(iv) A is a finite set of arcs.

(v) N is a node function. It is defined from A to $P \times T \cup T \times P$.

(vi) C is a color function. It is defined from P to Σ.

(vii) Con is a condition function. It evaluates either the rule condition if $t \in T_{rule}$ or it evaluates the time interval when $t \in T_{comp}$.

(viii) Action is an action function. It creates tokens according to action rules.

(ix) D is a time interval function.

(x) τ is a timestamp function.

(xi) I is an initialization function.

4. ECAPNSIM

Active rules development is an activity that needs to be performed carefully. Nowadays, there are few systems (Schlesinger and Lörincze 1997) which perform the analysis and debug the ECA rule base. Most of commercial ADBs (Hanson 1996), (Widom 1996) provide a syntax to ECA rule definition, however static analysis of ECA rules cannot be performed inside these systems and the ECA rule definition is only in a text way.

ECAPNSim (ECA Petri Nets Simulator) was developed under MAC OS X Server in Java. Taking advantage of Java portability, ECAPNSim can be executed in different operating systems. As an engine of ECA rules, ECAPNSim can be connected with any relational database systems such as Postgres, MS Access, Oracle, and Visual Fox Pro.

4.1 ECAPNSim architecture

ECAPNSim architecture consists of two building blocks: ECAPNSim Kernel and ECAPNSim tools environment (figure 1). ECAPNSim Kernel provides active functionality to passive database. it consists of CCPN Rule Manager, CCPN rule base, Composite Event Detector, and Rule Execution Component. ECAPNSim tools environment has a set of tools used by the ECA rule developers. Tools environment is composed by ECA rule editor, analyzer of no-termination problem, converter of ECA rules to CCPN, CCPN visualizer/editor and explanation components, termination analyzer and runtime tools.

ECAPNSim offers two modalities. In Simulation mode, users can simulate the behavior of the ECA rule base modeled by depositing tokens into the CCPN manually. And, in Real mode, the CCPN is executed by state modification of the connecting DBS.

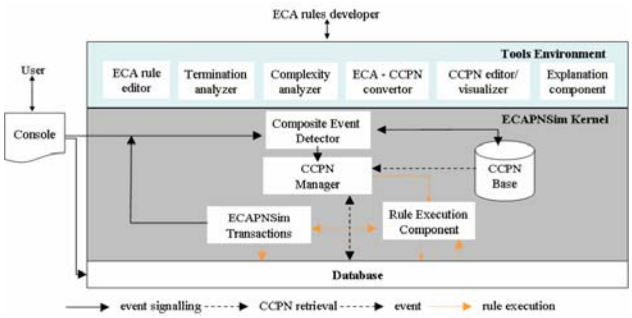


Figure 1: ECAPNSim architecture.

4.2 ECAPNSim Design

ECAPNSim offers a graphical and visual interface to represent ECA rule bases by CCPN model. Like any PN editor, ECAPNSim simulates the behavior of ECA rules by executing the CCPN model. Meanwhile simulation is running, problems like no termination and confluence can be observed obviously in the CCPN, hence ECA rule developer can modify the rule base to improve it.

The core of ECAPNSim is CCPN models. ECAPNSim contains a module to generate a CCPN structure from an ECA rule base definition written in a text file automatically. Or a CCPN model can be edited directly from a ECAPNSim user.

ECAPNSim supports CCPN design and edition from an ECA rule base, which can be moved to another position in the visualization panel. Moreover, because of there are large ECA rule bases, ECAPNSim will generate large CCPN structures, so it has zoom buttons to either increase or decrease the CCPN size. Simulation speed can be controlled through a slide. Finally, the graphical interface has tools and icons to edit a CCPN, simulate a CCPN behavior, and CCPN file management. (figure 2).

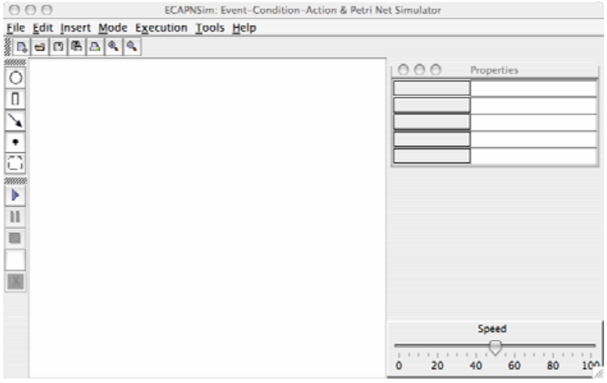


Figure 2: ECAPNSim environment.

4.3 Incorporation of distribution functions

ECAPNSim was enhanced with the addition of distribution functions, which are useful to simulate and to analyze the event occurrence in an active rule base.

Distribution functions which are available in ECAPNSim are beta, binomial, Cauchy, chi square, exponential, gamma, geometric, uniform, and weibull, among others. The use of this set of functions depends on the active rule base that will be simulated.

Each place in the CCPN has the property for the definition of a distribution function, according to the frequency of the event occurrence. The values for the functions can be determined by a statistical analysis of the data about the real occurrences of the events that fire ECA rules.

Random values which are generated by distribution functions are used as inter arrival time of events in ECAPNSim. Each time an event occurrence is simulated, a token with information about that event is created, and it is putted into the place that represents the corresponding event. Hence, the token game animation is started and ECA rule developer can detect inconsistencies in ECA rule set.

5. CASE OF STUDY

In order to show the modeling of a base of active rules as a CCPN, four ECA rules are converted into a CCPN, whose description is as follows:

Rule 01 : When an employee is inserted in the office DB and the production of employee's department is modified, if the production is greater than $900.00, then the employee's bonus is updated to $100.00.

Rule 02 : When either salary or bonus of an employee is modified, if the salary is increased by more than $200.00 or the bonus is increased by more than $50.00, then the employee's rank is increased too.

Rule 03 : When the employee's rank is updated, if rank value is greater than 15, then the employee's department budget is added with $1000.00

Rule 04 : When a department budget is modified, if the budget is greater than $20,000.00, then the department production is increased of 3%.

Definition of database tables needed to this rules are as follows:

DEP(TheDep,Production,Budget).

EMP(ItsDep,TheEmp,Salary,Bonus,Rank).

CCPN obtained from the rules listed above is showed in figure 3.

From the incidence matrix showed in figure 4, and from the CCPN picture showed in figure 3, it can be observed that there exists a cyclic path in the rule firing, however, the fact that there is a cyclic in the connections of CCPN elements is not a sufficient condition to ensure that there is a no termination problem. In order to imitate the behavior of ECA rule firing according to a real situation, event occurrences are modeled through the following functions:

E0 : Insert an employee. Uniform(4,4).

E1 : Update the production value of an department. Constant value of one day.

E3 : Update employee's salary. Uniform(315,50)

E4 : Update employee's bonus. Uniform(95,17)

E6 : Update employee's rank. Uniform(103,24)

E7 : Update department budget. Uniform(365,74)

The parameters are considered in days.

The evaluation of the conditional part has probability of occurrence of 50% for true and 50% for a false result.

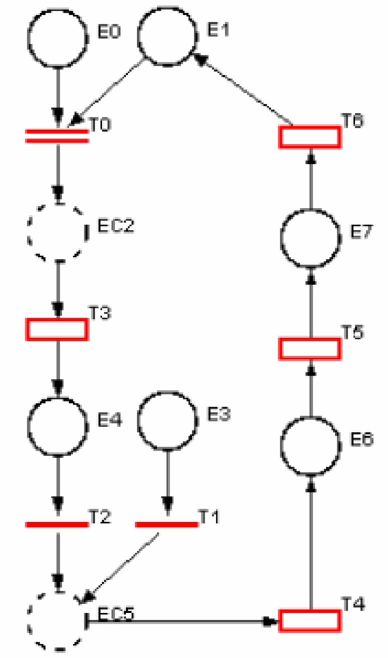


Figure 3. CCPN obtained from four ECA rules.

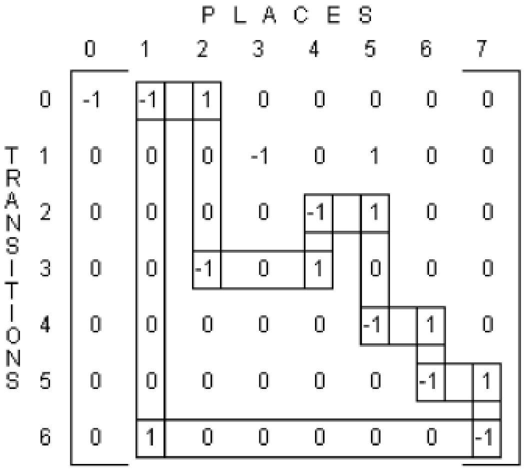
With these distribution functions assigned to each input place that represents event occurrence and running the simulation for a time corresponding to 10 years, the maximum value for the quantity of places visited by the same token was 10, i.e., the same token passed through ten places in the cyclic path, and the average value was 1.29 places visited; so this ECA rule base has the property of termination in its rule triggering.

6. CONCLUSION

Currently there are database management systems that support ECA rule definition by the use of "triggers", however "triggers" have several restrictions that limits the power that an active database must offer.

On the other side, there are research prototypes that support ECA rule definition, too; and they are more powerful because composite events such as conjunction, disjunction, etc., can de defined.

ECAPNSim is an interface that generates a CCPN from an ECA rule definition typed in the on-if-then form. It carries out the simulation of the CCPN behavior according to the event occurrence in a random way, which depends on the distribution function assigned.


PLACES
TRANSITIONS

	0	1	2	3	4	5	6	7
0	-1	-1	1	0	0	0	0	0
1	0	0	0	-1	0	1	0	0
2	0	0	0	0	-1	1	0	0
3	0	0	-1	0	1	0	0	0
4	0	0	0	0	0	-1	1	0
5	0	0	0	0	0	0	-1	1
6	0	1	0	0	0	0	0	-1

Figure 4. Incidence matrix of the CCPN showed in figure 3.

ECAPNSim has been improved with the addition of distribution functions in each place that denote an event occurrence.

A study area for active database is computer nets, where active behavior can be implemented in order to monitor traffic of LAN networks; an automatic reaction can be set by an active database system, instead of a human being monitoring. When suspicious events occur in the net, then ECA rules can be triggered and perform the corresponding action to maintain the stability in the net.

Another interesting area of application is in distributed database systems, where the development of ECA rules needs to consider the event occurrence in different servers.

There are several interesting areas where active databases are applied, mainly where systems need an automatic reaction and the response time must be immediate.

Nevertheless, ECA rule development implies that the ECA rule developer must be careful to avoid inconsistency states in the database system.

REFERENCES

Ceri, S., Fraternali, P., Paraboschi, S., Tanca, L., 1996. Active Rule Management in Chimera. In: Widom, J., Ceri, S., eds. *Active Database Systems: Triggers and Rules for Advanced Database Processing*. San Francisco CA :Morgan Kaufmann Publishers Inc., 151-176.

Collet, C., Coupaye, T., 1996. Composite Events in NAOS. *Proceedings of the 7th International Conference and Workshop on Database and Expert Systems Applications*. (*DEXA'96*). LNCS 1134, 244 - 253, September 9-13, Zurich, Switzerland.

Dayal, U., Blaustein, B., Buchmann, A., Chakravarthy, U., Hsu, M., Ledin, R., Mc-Carthy, D., Rosenthal, A., Sarin, S., Cary, M.J., Livny, M., Jauhari, R., 1988. The HiPAC Project: combining active database and timing constraints. *ACM SIGMOD RECORD*, 17 (1), 51- 70.

Gatziu, E., Ditrich, K.R., 1999. SAMOS. In: Paton, N. W., Ed. *Active Rules in Database Systems*. New York:Springer, 233-248.

Gehani, N., Jagadish, H.V., 1996. Active Database Facilities in Ode. In: Widom, J., Ceri, S., eds. *Active Database Systems: Triggers and Rules for Advanced Database Processing*. San Francisco CA :Morgan Kaufmann Publishers Inc., 207-232.

Hanson, E.N., 1996. The Design and Implementation of the Ariel Active Database Rule System. *IEEE Transactions on Knowledge and Data Engineering*. 8(1), 157-172.

Hursh, C.J., Hursch, J.L., 1991. *Oracle SQL Developer's Guide*. New York : McGraw-Hill.

Jensen, K., 1994. An Introduction to the Theoretical Aspects of Colored Petri Nets. *Lecture Notes in Computer Science: A Decade of Concurrency*, 803: 230- 272.

Lacy-Thompson, T., 1990. *INFORMIX-SQL, A tutorial and reference*. New Jersey : Prentice Hall.

Li, X., Medina-Marín, J., and Chapa, S.V., 2002. A Structural Model of ECA Rules in Active Database. *Lecture Notes in Artificial Intelligence, 2313* : 486-493.

Li, X., Medina-Marín, J., 2004. Composite Event Specification in Active Database Systems: A Petri Net Approach. *Proceedings of the IEEE International Conference on System, Man, and Cybernetics*, pp. 4885-4890. Oct 10-13, The Hague, The Netherlands.

McGoveran, D., Date, C.J., 1992. *A guide to SYBASE and SQL Server : a user's guide to the SYBASE product*, Boston : Addison-Wesley.

Medina-Marín, J., Li, X., 2005a. An Active rule base Simulator based on Petri Nets. *Proceedings of the the Third International Workshop on Modelling, Simulation, Verification and Validation of Enterprise Information Systems MSVVEIS-2005*, pp. 96-101. May 24-28, Miami, USA.

Medina-Marín, J., 2005b. *Desarrollo de reglas ECA, un enfoque de red de Petri*, Thesis (PhD). CINVESTAV-IPN, México.

Paton, N.W., Diaz, O., 1999. Active Database Systems. *ACM Computing Surveys*, 31 (1), 64- 103.

Schlesinger, M., Lörincze, G., 1997. Rule modelling and simulation in ALFRED, *Proceedings of the 3rd. International workshop on Rules in Database Systems (RIDS'97)* (or LNCS 1312), pp. 83-99. June 26-28, Skövde, Sweden.

Silberschatz, A., , Korth, H.F., Sudarshan, S., 1999. *Database System Concepts*. *Third Ed.* New York*:* McGraw-Hill.

Stonebraker, M., Kemmintz, G., 1991. The POSTGRES Next-Generation Database Management System. *Communications of the ACM*, 34(10), 78-92.

Widom, J., 1996. The Starburst Active Database Rule System. *IEEE Transactions on Knowledge and Data Engineering*, 8(4), 583-595.

AUTHORS BIOGRAPHY

Joselito Medina-Marín. He received the M.S. and Ph.D. degrees in electrical engineering from the Research and Advanced Studies Center of the National Polytechnic Institute at Mexico, in 2002 and 2005, respectively. He is presently a Professor of the Advanced Research in Industrial Engineering Center at the Autonomous University of Hidalgo State at Pachuca, Hidalgo, México. His current research interests include Petri net theory and its applications, active databases, simulation, and programming languages.

Xiaoou Li. Received the B.S. and Ph.D. degrees in applied mathematics and electrical engineering from Northeastern University, Shenyang, China, in 1991 and 1995, respectively. From 1995 to 1997, she was a Lecturer of electrical engineering with the Department of Automatic Control, Northeastern University. From 1998 to 1999, she was an Associate Professor of computer science with Centro de Instrumentos, Universidad Nacional Autónoma de México, México City, México. Since 2000, she has been a Professor of computer science at Departamento de Computación, Centro de Investigación y de Estudios Avanzados del Instituto Politécnico Nacional (CINVESTAV-IPN), México, City. Her research interests incluye Petri net theory and application, neural networks, information systems, data mining and system modelling and simulation.

José Ramón Corona-Armenta. He received the B.S. in Civil Engineering from Instituto Tecnológico de Pachuca in 1993, the M.S. in Engineering from the Universidad Nacional Autónoma de México in 1996, and the Ph.D. degree in Industrial Systems from Institut National Polytechnique de Lorraine in 2005. Since 2005 he has been a Professor of the Advanced Research in Industrial Engineering Center at the Autonomous

University of Hidalgo State at Pachuca, Hidalgo, México.

Marco Antonio Montufar-Benítez. He received the B.S. in Geophysical Engineering from Universidad Nacional Autónoma de México in 1985, and the M.S. in Operations Research from the Universidad Nacional Autónoma de México in 1990. Since 2000 he has been a Professor of the Advanced Research in Industrial Engineering Center at the Autonomous University of Hidalgo State at Pachuca, Hidalgo, México.

Oscar Montaño-Arango. He received the M.S. in Planning from Universidad Nacional Autónoma de México in 2000, and the Ph.D. degree Planning Systems from the Universidad Nacional Autónoma de México in 2007. Since 2006 he has been a Professor of the Advanced Research in Industrial Engineering Center at the Autonomous University of Hidalgo State at Pachuca, Hidalgo, México.

Aurora Pérez-Rojas. She received the B.S. in Industrial Engineering from Polytechnic Superior Institute José Antonio Echeverría (ISPJAE) in 1971, the M.S. in Automatic Systems from ISPJAE in 1978, and the Ph.D. degree in Technique Science from ISPJAE in 1987. Since 2006 she has been a Professor of the Advanced Research in Industrial Engineering Center at the Autonomous University of Hidalgo State at Pachuca, Hidalgo, México.

DEGOMS, A SYSTEMATIC WAY FOR TASK MODELLING AND SIMULATION

Ali Mroue$^{(a)}$**, Jean Caussanel**$^{(b)}$

(a)(b)Laboratory of sciences of informations and of systems
LSIS UMR 6168 University of Paul Cezanne, Aix-Marseille III
Av. escadrille de Normandie Niemen 13397 Marseille Cedex 20, France

(a)ali.mroue@lsis.org,, (b) jean.caussanel@lsis.org

ABSTRACT

If the representation of tasks can now rely on many modeling formalisms, few of them offer simulation solution for controlling the consistency of the modeled task. Most often this control can be done by a human agent working on the built model.

As part of a project of modeling and simulation of the human operator, we propose a process and a platform offering an opportunity for those who have to design applications with the known constraints of current developments. This paper describes our approach which uses GOMSL formalism, widely used in the world of HCI, as a model source. GOMSL is very suited for the description of computer task but it doesn't have a real simulator. The aim of this work was therefore to establish a semi-automatic re-expression of GOMSL models to DEVS model that can be immediately simulated (a set named DeGOMS).

Keywords: Task Model Simulation, HCI, DEVS, GDEVS.

1. INTRODUCTION

The Task models in computer science are useful for gathering and organizing the need for the user. They are normally used in the design phase of software to evaluate in advance, test and improve the design of the latter.

Few are the models that describe the real aspect of the user, they are normally based on descriptions of prescribed tasks (Tricot and Nanard 1997), which considers that the user masters the use of application. Even fewer are environments that offer a simulation solution for these task models in order to assess their consistency or their usability.

In addition the improvement of realism of this task models that we will not discuss it in this paper, also the ability to simulate the produced models are considered as an important way for verifying these models.

Our contribution to this field will therefore be as a first step, to design and develop such a platform for testing task models.

We have chose GOMS (Card, Newell and Moran 1983) as a task modeling formalism, which is used for modeling tasks from the view-point of user/system, specially in computer task modeling.

GOMS models can generally be used for the evaluation of interfaces (predicting execution and learning time).

GOMS models are widely used to evaluate and test at a lower cost, interfaces during the design phase. However, GOMS models are designed as models of representation and not as simulation models.

It is therefore necessary to adapt GOMS for simulation or be able to express it in a simulated formalism.

For this reasons we chose the DEVS formalism (Zeigler 1984), which offers a good level of expressiveness, a very good level of scalability, while ensuring formal consistency of the built models.

The equivalent of GOMS model in DEVS (which we called DeGOMS) is a generic level so that any GOMS model can be expressed in DEVS and then simulated. In this theoretical transformation approach, certain choices have been taken which can affect GOMS model (extending it).

2. GOMS MODEL

The acronym GOMS stands for Goals, Operators, Methods, and Selection Rules GOMS (Card, Newell and Moran 1983; John and Kieras 1996). GOMS is a behavior description model, that lets model the behavior at different levels of abstraction, from task level to physical actions.

GOMS uses as a starting point the Model Human Processor principle of rationality that attempts to model and predict user behavior. Its essential contribution is a formal structure that allows organizing the design process.

The design method that induces GOMS is done on two axes:

- In the analysis of task (since determines the behavior).
- In the predictive evaluation of user behavior in the task.

2.1. GOMS Element

GOMS is an approach of modeling human computer interaction.

GOMS models consist of descriptions of the methods required to accomplish a specific goal.

Methods are a sequence of operators and sub-goals to achieve a goal. If there exists more than one method to accomplish a goal then selection rules are used to choose which method to use.

- Goals are tasks the system's user wants to accomplish. For example, "Create Folder, Delete Word". A goal can have a hierarchical structure; this means that the achievement of a goal may require accomplishing one or more sub-goals.
- Operators are actions allowed by the software or actions that user are executing. An operator is an atomic level action that can't be composed, and it's characterized by its execution time. The execution of the operators causes change in the mental state of the users or in the environment state. There are two types of operators mental and physical, For example, the operators “press enter”, “point to the word”, etc. are physical operators. The model also includes mental operators, such as “thinking”, etc.
- Methods refer to the process that allows one to accomplish a goal. Methods are possible sequences of operators and sub-goals used to accomplish a goal. For example the goal of logging into web-mail can be represented as:

    1. Connect_to_the_webmail_provider
    2. Type username
    3. Type password
    4. Press Login

- Selection Rules are used when there exists more than one method that can accomplish the same goal. They are rules used by user to choose which of methods to use.
  A rule has the form:
  If <condition>
  Then use the method M;

GOMS has been used in many applications:
* Telephone operator (CPM-GOMS)
* CAD systems (NGOMSL)
* Text editing using the mouse (KLM)

2.2. GOMS Variation

There are four different models of GOMS: CMN-GOMS, KLM, NGOMSL and CPM-GOMS.

CMN-GOMS stands for Card, Moran and Newell GOMS, is the original version of the GOMS technique in human computer interaction. This technique requires a strict goal-method-operation-selection rules structure.

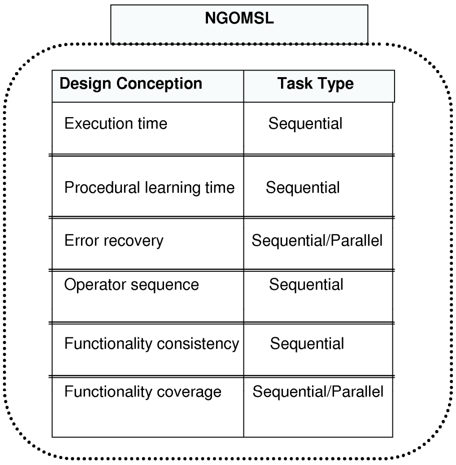
KLM is a simplified variant of CMN-GOMS, it does not use goals, methods, or selection rules only simple “keystroke-level operators”.

NGOMSL stands for Natural GOMS Language, developed by David Kieras (Kieras 2006). An NGOMSL model is in program form, and provides predictions of operator sequences, execution time, and time required to learn the methods. Like CMN-GOMS, NGOMSL models explicitly represent the goal structure, and thus can represent high-level goals like collaboratively writing a research paper" (John and Kieras 1996).

CPM-GOMS stands for Cognitive, Perceptual, and Motor and the project planning technique Critical Path Method. CPM-GOMS was developed in 1988 by Bonnie John (John and Kieras 1996). CPM-GOMS does not make the assumption that operators are performed serially, and hence it can model the multitasking behavior that can be exhibited by experienced users. The technique is also based directly on the model human processor a simplified model of human response.

In this paper we will introduce the equivalent of the NGOMSL in DEVS. Consequently we will details the NGOMSL of GOMS.

3. NGOMSL



NGOMSL

Design Conception	Task Type
Execution time	Sequential
Procedural learning time	Sequential
Error recovery	Sequential/Parallel
Operator sequence	Sequential
Functionality consistency	Sequential
Functionality coverage	Sequential/Parallel

Figure 1: NGOMSL design conception and task type capability

GOMS is a group of models developed by Card (Card, Newell and Moran 1983) and his colleagues at Xerox to predict the time needed to accomplish cognitive activities using a computer system. The model is designed to provide approximations for the duration of the task. The set of GOMS models assume that the user is quite familiar with the task and that the primary human limitations are cognitive (thought), not physiological (aerobic capacity, muscle strength, etc…).

The key of GOMS analysis is the decomposition of the task. As we mentioned above there are four of commonly GOMS implementations concepts CMN-GOMS, KLM, NGOMSL, and CPM-GOMS for more information see (John and Kieras 1996). We chose to work with NGOMSL, since it can be used in most types of tasks and information design (Kieras 2006) (Figure 1).

NGOMSL is a structured natural language used to represent user methods and selection rules.

NGOMSL models have an explicit representation of the user and his methods, which are supposed to be strictly sequential form and hierarchical. NGOMSL is based on the cognitive modeling of human-machine interaction.

In this paper we present a transformation of NGOMSL to DEVS model which is based on the definition of GOMSL (Kieras 2006) which is an executable representation of NGOMSL.

GOMSL can be treated and executed using a simulation tool named "GLEAN". This tool allows predicting task time and the performance of the user (learning time).

GOMSL syntax is comparable to that of a procedural programming language. As other GOMS models, GOMSL defines the task as of a method composed from steps. Each step is generally composed of one or more operators.

All data in GOMSL are generally represented as objects with properties and values (See LTM_item in Figure 2).

```
LTM_item: Copy_Command
            Name is Copy.
            Containing_Menu is Edit.
            Menu_Item_Label is Copy.
            Accelerator_Key is COMMAND-C.

Method_for_goal: Select Word
            Step 1. Look_for_object_whose Content is Text_selection of
            <current_task> and_store_under   <target>.
            Step 2. Point_to <target>; Delete <target>.
            Step 3. Double_click mouse_button.
            Step 4. Verify "correct text is selected".
            Step 5. Return_with_goal_accomplished.

Method_for_goal: Select Arbitrary_text
            Step 1. Look_for_object_whose  Content is Text_selection_start of
            <current_task> and_store_under <target>.
            Step 2. Point_to <target>.
            Step 3. Hold_down mouse_button.
            Step 4. Look_for_object_whose Content is Text_selection_end of
            <current_task>
            and_store_under <target>.
            Step 5. Point_to <target>; Delete <target>.
            Step 6. Release mouse_button.
            Step 7. Verify "correct text is selected".
            Step 8. Return_with_goal_accomplished.
```

Figure 2: Part of GOMSL methods and object definition

There are several kinds of data used in GOMSL. There is data that represent the knowledge of the users, data that represent the system and data that represent the task list.

The Part that contains data representing the knowledge of user are used and defined in:

Working Memory: Represents the working memory of the user; it has two partitions: Object Store and Tag Store...

- Object Store is a store that contain objects that are currently "in focus" in the user working memory. The Object Store can contains at the same time one Visual Object, one Long Term Memory Item (Object), one Task item (Object) and many Auditory Object with a condition. An auditory object is characterized by a decay time, after it the auditory object will decay.
- Tag store represents the conventional human working memory which can contains values associated with tags (ie: Keyword=Html, the tag is keyword which has HTML as value)

GomsL define many operators that deals with memory (storing value and getting values etc…).

Long Term Memory: Represents the Long Term memory of the user, which contains objects with property values. The content of the LTM is static being specified before the simulation of the model.

The data concerning the information about the system is stored in a memory similar to structure of LTM, and it contains system visual and auditory objects that will be needed by the user to execute task (look at a specific object etc…).

Task memory: This is used to represent the information available to the user about the task.

3.1. Operators

Operators are actions that the user performs. Much kind of operators are defined in GOMSL:

3.1.1. External Operators

They are observable actions used by the user for exchanging information with the system or with other humans. Normally external operators depend of the system and the task. Below is a list of external operators classified by their type, the time and the definition of some of these operators are based on the physical and some of the mental operators used in the Keystroke-Level Model (Kieras 2006).

Manual Operator:

- KeyStrock Keyname (Stroke a key on the keyboard. Estimated execution time 280ms).
- Click mouse_button (clicking the mouse button. Estimated execution time 200 ms).
- Hold_down mouse button (press and continue pressing the mouse button. 100ms)
- Etc…

Visual Operator:

- Look_for_object_whose property is value, … and_store_under <tag> (its a Mental operator that searches on the system for a visual object that whose specified properties have the specified values, and stores its symbolic name in the tag store section of the WM, and put the object in focus in the visual part of store object. After the execution of this operator all the properties of the object become available. Execution time is estimated to 1200 ms.

- Look_at object_name (Means looking at an object that has already been identified with a Look_for… operator etc… 200ms)
- Etc…

Etc…

3.1.2. Mental Operators

Mental operators are user internal action, they are not observable or hypothetical, and they are inferred by the analyst or the theorist.

Some of these operators are built in; they are primitive operators corresponding to the mechanism of cognitive processor. These operators include actions like taking decisions; store a value in tag store, search for information in the Long Term Memory, get information about the Task, etc… Other operators are defined by the analyst in order to represent complex mental activity such as "Verify", "Think Of" …

Memory storage and retrieval:

- Store Value under <tag> (Store a value in the tag store under the label tag).
- Delete <tag> (Delete the value stored under the label tag)
- Etc...

flow of control:

- Accomplish_Goal: Goal_name (Accomplish a goal, the goal will be considered as a sub-goal)
- Decide: Conditional; Conditional; …; else-form (This operator let us taking decision. It contains one or more If-Then conditionals and at most one Else form)
- Etc…

Etc…

4. DEVS FORMALISM AND GDEVS

DEVS (Zeigler 1984) is a formalism used to represent Discrete Event System Specifications; it can represent complex system in an effective way. DEVS model is a powerful simulation model. It is a modular formalism for deterministic and causal systems. It allows for component-based design of complex systems. Several specific platforms for DEVS models simulation can be found.

A DEVS model may contain two kinds of DEVS components: Atomic DEVS and Coupled DEVS. An Atomic DEVS does not contain any component in it. It only has a mathematical specification of its behavior. A Coupled DEVS is a modular composition of one or more Atomic and Coupled DEVS.

An atomic DEVS model has the following structure:

$D = \langle XD, YD, SD, \delta extD, \delta intD , \lambda D, taD \rangle$

Where:

XD: is the set of the input ports and values

YD: is the set of the output ports and values

SD: the sequential state set

$\delta extD$: $Q \times X \rightarrow S$, is the external transition function where $Q = (s, e) \rightarrow s \in S, 0 \leq e \leq ta(s)$ is the total state set.

$\delta intD$: $S \rightarrow S$, is the internal transition function

λD: $S \rightarrow Y$, is the output function

taD: $S \rightarrow R+0, \infty$ (non-negative real), is the time advance function.

A coupled model, also called network of models, has the following structure:

$N = \langle X, Y, D, \{Md / d \in D\}, EIC, EOC, IC \rangle$

X and Y definitions are identical to XD and YD of an atomic model. The inputs and outputs are made up of ports. Each port can take values and has its own field of values.

$X = \{ (\rho,\mu) / \rho \in IPorts, \mu \in X\rho \}$

$Y = \{ (\rho,\mu) / \rho \in OPorts, \mu \in Y\rho \}$

D is the set of the model names involved in the coupled model.

Md is a DEVS model. The variables representing the inputs and the outputs of the model will be indexed by the model identifier. Hence the following notation:

$Md = \langle Xd, Yd, Sd, \delta extd, \delta intd, \lambda d, tad \rangle$

The inputs and the outputs of the coupled model are connected to the inputs and outputs of the included models.

$EIC = \{((N, a), (d, b))/a \in IPorts, b \in IPortsd\}$

The set of the coupled model input ports ipN associated with the input ports ipd of the models D are the components of the coupled model.

There is the same situation for the output ports.

$EOC = \{((N, a), (d, b))/a \in OPorts, b \in OPortsd\}$

Inside the coupled model, the outputs of a model can be coupled with the inputs of the other models. An output of a Model cannot be coupled with one of its inputs.

$IC = \{((i, a), (j, b))/i,j \in D, i, j,a \in OPortsi, b \in IPorts j \}$

GDEVS is an acronym for"Generalized Discrete Event System Specification" (Giambiasi, Escude and Ghosh 2001) which is a model which generalizes the concept of discrete event modeling. GDEVS defines abstraction of signal with piece wise polynomial trajectory. Thus, GDEVS defines event as a list of values. DEVS can be considered as a particular case of GDEVS, other saying DEVS is an order 0 GDEVS model.

5. GOMSL TO DEVS

GOMSL is a model of representation that is intended to be interpreted and verified by specialists in the field.

It lies at a high level of abstraction to facilitate its interpretation by humans but don't let us testing automatically the produced models.

We believe there is an interest to propose a simulation of represented tasks in order to obtain results or to ensure consistency, in particular when they have reached a certain level of complication.

The choice of target formalism in which the GOMSL representation will be translated is not critical as long as it ensures consistency syntactical and semantic of constructed models and let them be simulated directly.

For the reason of type of models, our choice fell on Discrete EVent System formalism DEVS simulation (Zeigler 1984; Giambiasi, Escude and Ghosh 2001).

In addition to the characteristics listed above, DEVS is generally having a very good level of scalability and expressiveness (Zeigler 1984).

In addition, we have now in our laboratory a DEVS simulation platform (lsisDME) through which we could test our results immediately.

5.1. Transformation

A GomsL model is equivalent to a coupled DEVS model.

During the transformation we separate the GOMS Methods from other models. So we create a model that represents the user, a model that represents the task, another that represents the system and final one that represents GOMS methods.

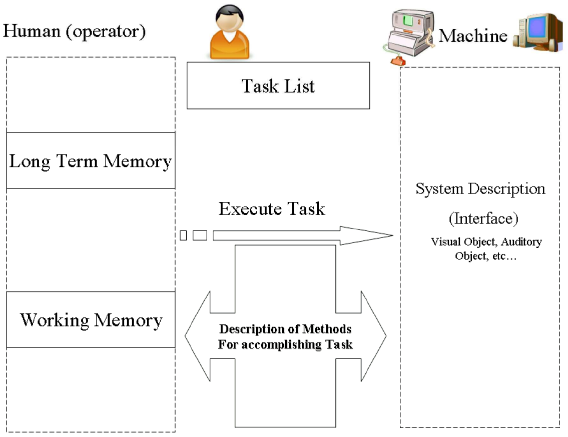


Figure 3: Decomposition of GOMS model in DEVS

This decomposition of model let us obtaining a modular architecture which will be useful so any change in any model will not affect other models moreover any model can be used alone (Figure 3).

The start point is the model representing Methods and Selection Rules. Each method is composed from a sequence of steps. Each step is composed from one or more operators. These operators can communicate with the other different models, in order to get value from memory or save value etc…

In this paper we will describe in detail working memory module used in GOMSL and its equivalent in DEVS (Figure 4). We will discuss then the description of a part of the module that represents “methods and selection rules”.

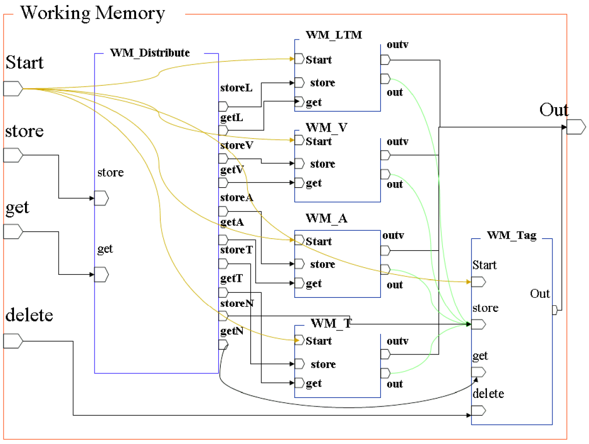


Figure 4: Working Memory GDEVS representation

5.1.1. Working Memory Model

Expressed according to the GDEVS formalism, the "Working Memory" can be defined as follows:

WM: <X, Y, M, EIC, EOC, IC, select> {
X = {Start[0], store[2], get[2], delete[0]};
Y = {Out[1]};
M = {WM_Distribute, WM_LTM, WM_V, WM_A, WM_T, WM_Tag};
EIC = {{Start, WM_LTM.Start}, {Start, WM_V.Start}, {Start, WM_A.Start}, {Start, WM_T.Start}, {Start, WM_Tag.Start},
{store,WM_Distribute.store},{get,WM_Distribute.get}
};
EOC = {{WM_LTM.outv, out}, {WM_V.outv, out}, {WM_A.outv, out}, {WM_T.outv, out},{WM_Tag.out, out}};
IC = {{WM Distribute.storeL, WM LTM.store},
{WM Distribute.getL, WM LTM.get},
{WM Distribute.storeV, WM V.store}, {WM Distribute.getV,WM V.get}, Etc. . . };

The Working Memory module is equivalent to a GDEVS coupled model, which is composed from six models: WM_Distribute, WM_LTM, WM_V, WM_A, WM_T, WM_Tag. It has 4 inputs ports (Start, Store, Get, Delete) and only one output port (Out).

WM_Distribute has as role to distribute every request to the proper model. For example, if WM_Distribute receives a request for retrieving a value from the visual memory part, so the WM_Distribute redirects the request to the proper model which is WM_V. This latter will receive the request and searches the value and sends it to the output.

Models "WM_LTM", "WM_V", "WM_T" are GDEVS atomic models that have a similar structure. Each of them represents a specific part of the Working

Memory, since, according to GOMSL, the working memory may contain various objects at the same time in different zone (visual area, LTM area, Task area). And each zone may contain a single object at the same time (when an object is present in the memory, all his properties become accessible in the methods).

An object in GOMSL is characterized by a name, and by properties with certain values. In DEVS an object is represented by a phase in which are defined state variables having values (These state variables represent the properties of the object).

As seen in the preceding parts, GOMSL defines that the working memory may contain at the same time a visual object, an object from the LTM and a Task object. By respecting this definition, we have created the models (WM_V, WM_LTM, WM_T).
Each of these models is equivalent to an atomic GDEVS model with the following structure:
WM_V = <Xd, Yd, Sd, δextd, δintd, λd, tad>

X = {Start[0], store[1], get[1]};
Y = {out[2],outv[0],message[0]} ;

There are five major states which are (init, receive, select, get, clear) and "n" other states. "n" is equal to the number of objects be stored in memory (Figure 5).

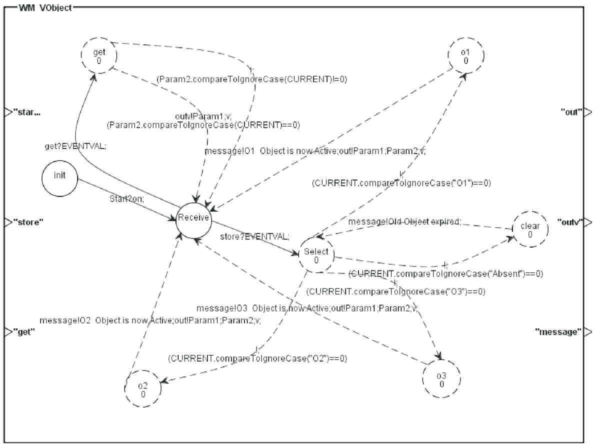

WM VObject
get
0
o1
0
(Param2.compareToIgnoreCase(CURRENT)!=0)
"star...
"out"
outv!Param1;v;
(Param2.compareToIgnoreCase(CURRENT)==0)
get?EVENTVAL;
message!O1 Object is now Active;out!Param1;Param2;v;
init
(CURRENT.compareToIgnoreCase("O1")==0)
Start?on;
"store"
Receive
message!Old Object expired;
clear
0
"outv"
store?EVENTVAL;
Select
0
(CURRENT.compareToIgnoreCase("Absent")==0)
(CURRENT.compareToIgnoreCase("O3")==0)
message!O3 Object is now Active;out!Param1;Param2;v;
message!O2 Object is now Active;out!Param1;Param2;v;
"get"
"message"
(CURRENT.compareToIgnoreCase("O2")==0)
o2
0
o3
0

Figure 5: Example of WM_V model

The init phase: Is the initial phase of the component model. It contains the initialization all state variables used in the model.

The Receive phase: This phase can receive two types of requests (storage and getting requests), and according to them it will go to the next phase. For example, if it receives a request to store a value the next phase will be the "Select" phase; else if it receives a get request next phase will be the "get" phase.

The Select Phase: This phase is used to select the memory object to be stored. In other words, after receiving a store request with a tag name and an object name. The select phase will select the object and make it active, what makes all its properties available to future get requests.
For example, if we receives a store (Tag1; O1), then we will go from the Select phase to the phase representing O1, after this the system will send to the output port "out" the values (tag;O1;v), which will be received by the WM_Tag in order save O1 in Tag …. Then the system returns to the receive phase and wait new request.

The Get Phase: This phase will search for a property value and sends it to the "outv" output port.

The clear phase: The role of this phase is to reset the values of all state variables. It will be used when we want to store a new object so the old object must be removed (all its properties must be cleared).

As we mentioned above the n other states represent the object in memory. When the system goes to one of these phases, all the state variables of this phase will be affected with values that represents properties values. After this the object will be active and all its properties will be available for get requests.

The models WM_LTM and WM_T have the same structure of the WM_V model, but n in these cases will be the number of LTM objects or TASK objects.

The WM_A model is different from the other models for the reason that working memory can contains many auditory objects in the same time, but in this case decay time will be taken in consideration. So object after a decay time will automatically be removed.

The WM_A is equivalent to a GDEVS coupled model, composed from n model, where n is the number of auditory objects.

Every auditory object is equivalent to an atomic GDEVS model with the following structure:
WM_V = <Xd, Yd, Sd, δextd, δintd, λd, tad>

X = {Start[0], store[1], get[1]};
Y = {out[2],outv[0],message[0]} ;

This model is composed of 6 phases (Figure 6) which are (init, get, receive, select, get1, get and a phase representing the object).

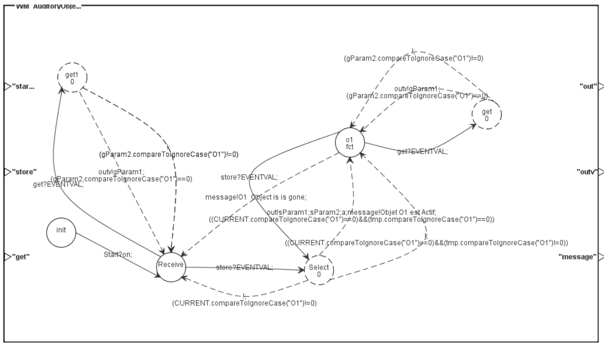

WM AuditoryObje...
(gParam2.compareToIgnoreCase("O1")!=0)
get1
0
"star...
"out"
outv!gParam1;
(gParam2.compareToIgnoreCase("O1")==0)
get
0
o1
fct
get?EVENTVAL;
(gParam2.compareToIgnoreCase("O1")!=0)
"store"
"outv"
outv!gParam1;
(gParam2.compareToIgnoreCase("O1")==0)
get?EVENTVAL;
store?EVENTVAL;
message!O1 Object is is gone;
out!sParam1;sParam2;a;message!Objet O1 est Actif;
((CURRENT.compareToIgnoreCase("O1")==0)&&(tmp.compareToIgnoreCase("O1")==0))
init
((CURRENT.compareToIgnoreCase("O1")==0)&&(tmp.compareToIgnoreCase("O1")!=0))
Start?on;
"get"
"message"
Receive
store?EVENTVAL;
Select
0
(CURRENT.compareToIgnoreCase("O1")!=0)

Figure 6: Example of WM_A model

The phases are similar to the ones used in WM_V model, with minor difference in the transition conditions and lifetime function. In other words the phase representing the object "O1" has a lifetime equal to the decay time of the object, after this time all the properties will not be accessible.

The model "WM_Tag" is an atomic GDEVS model representing the "Tag Store" part of the Working Memory.

Each object in the memory must be linked to a tag in the "Tag Store". So we can access objects by using tags. For this reason we can find in the working memory module that all object models are connected to the WM_Tag. As we mentioned above the WM_Tag (Figure 7) is equivalent to an atomic GDEVS model composed of five phases (init, tag, tag_store, tag_delete, tag_get).

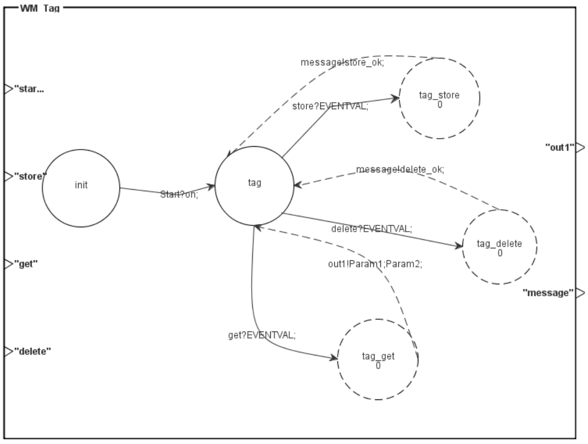


Figure 7: Example of WM_Tag model

The init phase is the initial phase; it contains the definition of all the tags used in the task as a state variables. In other words this phase is used to initialise the values of the state variables.

The tag phase is a waiting phase which will wait for an input event. This phase can deal with three type of events (delete, store, get).

The tag_store phase is used to store a value in a given tag. In other words, after receiving in the store input port a value such: store: current_task;aaa;n; this will be equivalent to current_task=aaa, and current_task_type=n; so in the phase tag_store we will assign value to the state variable.

The tag_get phase is used to retrieve the value from a variable (tag).

The tag_delete phase is used to remove the values from a variable (tag).

The input ports are X= {Start [0], store [2], get [0]};

The output ports are Y = {out [1], message [0]};

The definition of the input and the output port is similar to the one used in the above model.

More generally the working memory module receives at the start of simulation in the port Start an activation signal (on), which puts all the components of the memory in a waiting phase. Then it can either receive requests for storage ("store" input port) or retrieving data (“get” input port) or requests to delete a tag in the WM_tag ("delete" input port).

For example: if the WM receives Store: (query; html; t), the "WM_distribute" analyses all the entries and chooses model to activate depending on the type of application (in this case is part WM_T indicated by the “t” in the input) and then the model will send (query; html) to the port StoreT. Then, the WM_T model receives on its input port store the values (query; html). So the WM_T will search for the object named HTML and makes it 'active' (all properties of this object will be available...). After that, WM_T sends the values (query; html; t) to the output port "out". Finally, WM_Tag receives on the input port values (query; html; t) and stores in the tag (state variable) name "query" the value "html", and stores also in the state variable “query_type” the value “t” which means "task object”.

5.1.2. GOMS Module

This module represents in general goal, operators, methods and selection rules.

The "Goal" is equivalent to a GDEVS coupled model, composed from a “Selection Rules” model and methods models. The name of the model is equivalent to the goal name.

The model have 5 input ports (Start, memin, taskin, systemin, inaccomplish) and 8 output ports (memstore, memget, memdelete, taskcmd, systemget, LTMget, outaccomplish, accomplished).

Input Ports:

Start: used to indicate the start of execution of the model.

Memin: is the port used to receive values from the working memory.

Systemin : This port is used to receive values from the System.

Taskin: used to receive values from the Task description model.

Inaccomplish: is used to receive the end of execution of another goal called from this goal.

Output Ports:

Memstore: used to send request to the WM for storing values or object.

Memget: used to send request to the WM for retrieving values (object properties values, tag values, etc...)

Memdelete: send a delete request to the WM (The WM_tag exactly) for deleting a tag.

Taskcmd: used to send command to the Task description model (modifying the task list, getting values …)

Outaccomplish: used to send request for executing a goal.

Etc.

The simulation starts with the Selections Rules model. This latter is defined in GOMSL in the form of simple conditions (if then .....). The Selection Rules model is equivalent to an atomic GDEVS model. It has two input ports (memin and Start) and two output ports (memget, out). The port "out" in this case is used to transmit the name of the method selected (Figure 8).

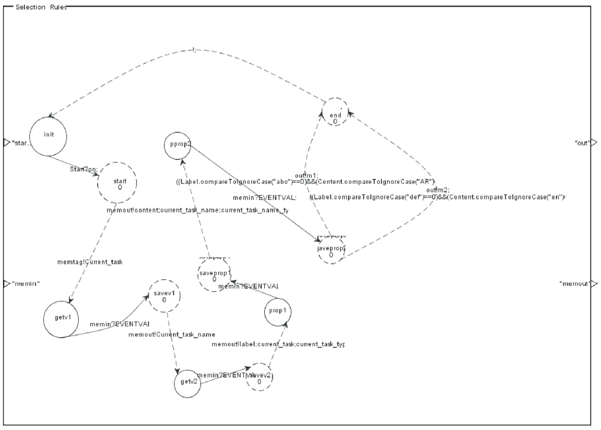

Selection Rules
"star..
"memin"
"out"
"memout"
init
start
0
Start?on;
end
0
pprop2
saveprop1
0
saveprop2
0
getv1
savev1
0
getv2
savev2
0
prop1
memout!content;current_task_name;current_task_name_ty
memtag!Current_task
memin?EVENTVAL
memout!Current_task_name
memout!label;current_task;current_task_typ
memin?EVENTVAL
memin?EVENTVAL;
out!m1;
((Label.compareToIgnoreCase("abc")==0)&&(Content.compareToIgnoreCase("AR")
out!m2;
((Label.compareToIgnoreCase("def")==0)&&(Content.compareToIgnoreCase("en")

Figure 8: Example of Selection Rules Model

This model is composed of three main phases (init, start and end), plus 2*n phases, with n equal to the number of values we need to retrieve from the “WM”. The multiplication by 2 is due to the reason that we have to one phase for requesting values from the WM and another for receiving the value.

First of all this model sends requests to the WM in order to get all needed values, then these values will be compared and according to the result of comparisons the name of proper method to execute will be sent to the output port (out).

The phase that is linked directly to the phase "End" will send the name of the method to execute (according to the condition on transitions. . .)

A Method in GOMSL is equivalent to a coupled GDEVS model composed from n atomic GDEVS model. The “n” Atomic GDEVS model represents the steps of the NGOMSL model.

A Step in GOMSL can be composed from one or more operators.

An Operator in DEVS is equivalent to one or more phases according to the type of operator. For example: An external operator such as: Click mouse_button is normally equivalent to one phase, the life time of the phase is equal to the execution time of the operator. If this operator needs an access to the memory, then we add 2 other phases in order to send the request to get the value and save it.

Another example for the mental operator such as think is equivalent to one phase etc...

6. DEGOMS ADVANTAGE

The translation of GOMSL into DEVS has extended the features of GOMSL model.

1. To simplify, Gomsl considers that the auditory memory can contain all auditory objects simultaneously. In DeGOMS the contrary, we represent this part of the memory as it is defined, which is to take into consideration the decay time
2. In GomsL, one can not easily add operators. Each added operator needs recompilation and reconfiguration of the existing simulation tools (which is "Glean"). While in DeGOMS, we introduced a generic concept the definition of the operator (example: 1-Request for a value from the memory needs two phases 2-The running time of the operator is equivalent to the life time of the phase, and so on).
3. The modular architecture of DeGOMS allows us to use each module alone, in other applications etc… In the case of modification of a module this will affect only the modified module and not all the modules. In GomsL and Glean this is not possible; since these models are not defined formally (we mostly change the code etc).
4. DeGOMS can be easily expanded by adding models that already exist or creating new models that can be simulated directly. In GomsL/Glean any change is quite complex, and requires coding and recompilation of Glean.
5. Etc.

7. RESULTS

The first results of simulations of DEGOMS models, show the proper functioning of each module taken separately (memory module, method, and so on…)

The simulation of a simple model for the management of files in Macintosh System validates the model and gives good results. It gave us the possibility to obtain the same result of an existing Glean simulator (ex: Accomplishing task time: will be the date of the end of the simulation (Last Event). The advantage of DeGOMS models is that they provide a very large modelling flexibility (easily change models, adding new operators, and adding new methods) also they provide us and the possibility of simulation directly.
This flexibility is not found in an approach as GLEAN (Kieras 2006) which it is a runtime environment model for GOMSL more than a true simulation model.

8. CONCLUSION

The simulation of the operator in its task is challenging, especially when the modeled system includes a human interaction. In the design of computer applications this type of approach allows us to minimize the cost and time of the design while improving the overall ergonomics applications. We designed DeGOMS and developed a DEVS formalism representing GOMSL models and an environment which allows the simulation of produced models.

In this work we succeeded in finding a new simulation tool for the NGOMSL model, which is effective and gives good results. This transformation into DEVS has many advantages, first we have obtained an effective simulation tool for the NGOMSL model, and secondly we can easily interpret these models and extend them in order to add many other modules. And for now we can also do the simulation in interactive mode, and implement the core of the DEVS simulator in a user training applications. The current prospects relate to model the task of information retrieval on the web and simulating it.

9. REFERENCES

Tricot, A. and Nanard, J. 1997. *Un point sur la modélisation des tâches de recherche d'informations dans le domaine des hypermédias*, in Hypertextes et Hypermedia J.P. Balpe, et al.,Editors. Hermes, 35-56.

Card, S. K.; Newell, A. and Moran, T. P. 1983. *The Psychology of Human-Computer Interaction.* Lawrence Erlbaum Associates, Inc. Mahwah, NJ, USA.

Kieras, D.E. 2006. *A Guide to GOMS Model Usability Evaluation using GOMSL and GLEAN4*. University of Michigan

John B. E. and Kieras D. E. 1996. *The GOMS family of user interface analysis techniques: comparison and contrast.* ACM Transactions on Computer-Human Interaction, 3(4):320–351.

Zeigler B. P. 1984. *Theory of Modelling and Simulation.* Krieger Publishing Co., Inc. Melbourne, FL, USA.

Giambiasi, N. Escude, B. Ghosh, S. 2001. *G-DEVS A Generalized Discrete Event Specification for Accurate Modeling of Dynamic Systems in:* Autonomous Decentralized Systems. Proceedings. 5th International Symposium on.

NET CENTRIC MODELLING AND SIMULATION USING ACTORDEVS

Franco Cicirelli[(a)], Angelo Furfaro[(b)], Andrea Giordano[(c)], Libero Nigro[(d)]

[(a) (b) (c) (d)] Laboratorio di Ingegneria del Software (www.lis.deis.unical.it)
Dipartimento di Elettronica Informatica e Sistemistica
Università della Calabria
87036 Rende (CS) – Italy

[(a)]f.cicirelli@deis.unical.it, [(b)]a.furfaro@deis.unical.it, [(c)]agiordano@deis.unical.it, [(d)]l.nigro@unical.it

ABSTRACT

The goal of the DEVS-World project is the development of a net-centric modelling and simulation (NCMS) infrastructure having the net as *the* computer, thus favouring different levels of interoperability among research groups operating world wide. This paper proposes an architecture based on web services for NCMS using ActorDEVS. ActorDEVS is a lean and efficient agent-based framework in Java supporting modelling of Parallel DEVS systems under both centralized and distributed simulation. ActorDEVS supports custom control engines. The paper discusses some architectural scenarios for wrapping ActorDEVS in the DEVS-World infrastructure, opening to interoperability with other DEVS or (possibly) non-DEVS systems. The proposal clearly separates model and simulation concerns. An entire model is partitioned among a number of simulation nodes with web services, in a case, which act as the transport layer for inter-node message exchanges. A global coordinator with a minimal interface of operations governs the "in-the-large" simulation aspects.

Keywords: M&S using the Internet, agent-based DEVS, web services, interoperability

1. INTRODUCTION

The DEVS-World project (DEVS-World 2007) aims at developing a world-wide standard platform for modelling and simulation (M&S), promoting collaborative research and experimentation in the engineering, i.e. design, evaluation, implementation, deployment and execution of complex, scalable, dynamic structure systems (Hu *et al.* 2005) belonging to diverse and significant problem domains like biology and bioinformatics, environment systems, traffic simulation etc.

The project has its strength in the use of DEVS (Zeigler *et al.* 2000) as the unifying M&S formalism and an exploitation of nowadays software technologies and middleware such as agents (Agha 1986, Wooldridge 2002, Cicirelli *et al.* 2007a) and services (Papazoglou and Georgakopulos 2003, Cicirelli *et al.* 2007c), which are a key for software interoperability. The main goal is enabling the exchange of both models and experiments among researchers and developers operating in academic or industry labs, thus favouring cooperation.

In this paper the ActorDEVS (Cicirelli *et al.* 2006, Cicirelli *et al.* 2007b, Cicirelli *et al.* 2008) framework is put under the perspective of DEVS-World in order to identify possible extensions and cooperation scenarios. ActorDEVS (see Fig. 1) is a lean and efficient agent-based framework in Java supporting modelling of Parallel DEVS systems under both centralized and distributed simulation. The approach clearly separates modelling from simulation concerns.

Both simulation and real-time execution modes are supported for model continuity which rests on the possibility of changing the control engine and ultimately the time notion regulating the evolution of the application. The approach is *control-centric*, in the sense that it allows customizing the control machine (see Fig. 1) which offers basic scheduling and dispatching message services to actor components.

Key factors underlying ActorDEVS are the adoption of actors (Agha 1986, Cicirelli *et al.* 2007d) as programming in-the-small building blocks, and of theatres (Cicirelli *et al.* 2007a) as programming in-the-large execution loci (see Fig. 1). Adopted actors are thread-less reactive objects which encapsulate an internal data state (which include *acquaintances*, i.e. known actors which can be contacted by messages), have a behaviour patterned as a finite state machine, and communicate to one another by asynchronous message passing. Actors can migrate dynamically from a theatre to another for reconfiguration purposes.

ActorDEVS is supported by a minimal API in Java. Typed input/output ports are mapped on to actor messages. Configuration operations correspond to updating receiver information in output ports, also during the runtime. More in general, changing actor's acquaintance network, a concept which is often referred to as link mobility, is a natural way to achieve model structure dynamism (Cicirelli *et al.* 2007a, Cicirelli *et al.* 2007d, Cicirelli *et al.* 2008). Good execution performance is ensured by having a DEVS model is flattened from the point of view of the simulation engine.

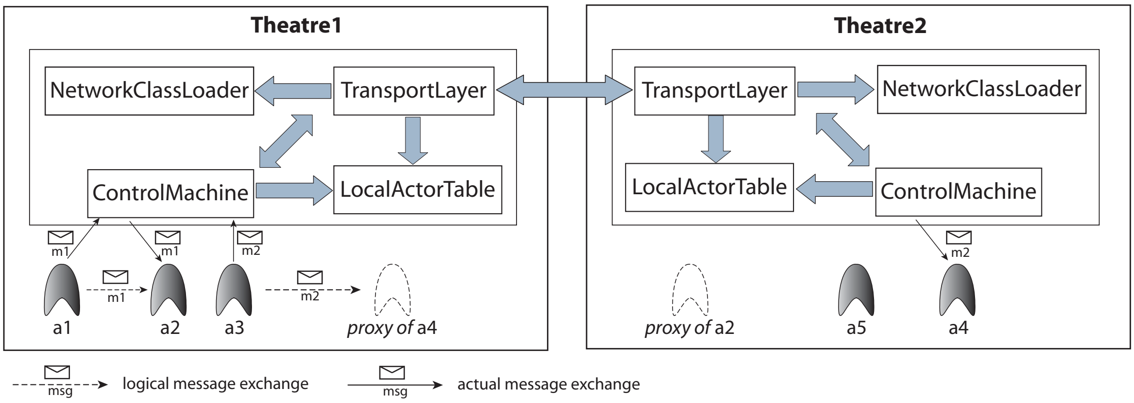


Figure 1: Actor/Theatre architecture for ActorDEVS

The paper is structured as follows. Subsequent section introduces the DEVS-World project and its objectives and key features. After that, a description is provided of how Theatre/ActorDEVS architecture can be extended for supporting NCMS. Last section discusses some issues relevant to a pragmatic use of the resultant architecture. Finally, future and on-going work is summarized.

2. AN OVERVIEW OF DEVS-WORLD

Novel in the DEVS-World project is the definition of a development methodology for supporting world-scale distributed *open* systems of systems M&S (DEVS-World 2007). Openness is a fundamental property which expands along different directions with different levels of integration and interoperability.

A first level of integration is relevant to model interoperability. Many different implementations of DEVS simulators currently exist, and usually each of them uses a built-in modelling language often tied to a specific programming language like Java or C++. To cope with this problem, specific conversion tools capable of translating a DEVS model from a language to another can be realized. A more general solution would be that of adopting emerging DEVS standard language such as DEVSML (DEVS-World 2007).

Another direction of integration concerns interoperability at architectural level. In (DEVS-World 2007) but also in (Mittal *et al.* 2008) the proposed world-wide architecture is aimed at harmonizing heterogeneous models based on special-case DEVS tools, programming languages and engines, through the use of Web Services and SOAP dependent messages and other DEVS concepts (ports, simulators, coordinator etc.). Web Services are viewed as a world-wide *glue* enabling interoperation through DEVS/SOA mechanisms, with WSDL used for web services interface specification.

Besides standardization of models and simulation infrastructure, the definition of a standard simulation protocol (Xiaolin and Zeigler 2008) is mandatory. The protocol (see Fig. 2) describes how a DEVS model should be simulated and how service/simulation engines should coordinate each other. Such a protocol opens also to a scenario in which both DEVS and non-DEVS simulators may (possibly) participate in a simulation.

CoreSimulatorInterface (see Fig. 2) is the common interface to simulators. The term “core” means “essential” in that as long as a simulator implements this interface, it can participate in a simulation driven by a DEVS coordinator. In the case of DEVS-simulators, the *CoupledSimulatorInterface* is considered. This interface extends the core interface by providing other functionalities e.g. for adding/removing couplings among DEVS models.

CoordinatorInterface must be implemented by the coordinator. The coordinator is in charge of synchronizing the activities of the various simulators guiding them through the simulation control cycle. Basic phases of the simulation cycle are shown in Table 1.

In handling simulation of hierarchical coupled models, a coordinator orchestrates a set of controlled simulators within it and, at the same time, can participate with peers in a coupled model above it. To allow such downward/upward facing interfaces, the *CoupledCoordinatorInterface* is introduced which extends both the *CoordinatorInterface* and the *CoupledSimulatorInterface*.

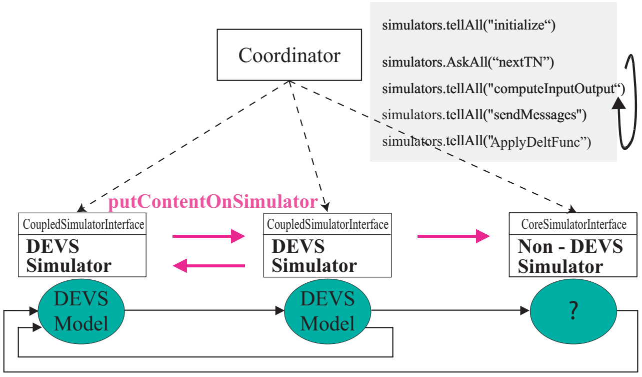

Coordinator
simulators.tellAll("initialize“)
simulators.AskAll(“nextTN”)
simulators.tellAll("computeInputOutput“)
simulators.tellAll("sendMessages")
simulators.tellAll("ApplyDeltFunc”)
putContentOnSimulator
CoupledSimulatorInterface
DEVS Simulator
CoupledSimulatorInterface
DEVS Simulator
CoreSimulatorInterface
Non - DEVS Simulator
DEVS Model
DEVS Model
?

Figure 2: Simulation protocol in a federation of DEVS and non-DEVS simulators

Table 1: Simulation cycle phases

Step	Description
nextTN	the coordinator requests that each simulator sends its time of next event and takes the minimum of the returned values to obtain the global time of next event
computeInputOutput	each simulator applies its *computeInputOutput* method to produce/gather an output that consists of a collection of *Contents* (i.e. port/value pairs)
sendMessages	each simulator partitions its output into messages intended for recipient simulators and sends these messages to these recipient simulators. Sending a message implies to call the recipient's *putContentOnSimulator* for any target simulator
applyDeltFunc	each simulator executes its *ApplyDeltFunc* method which computes the combined effect of the received messages and internal scheduling on its state. A side effect is in producing the time horizon gives back at the nextTN

3. WRAPPING ACTORDEVS IN DEVS-WORLD

This section highlights a service-based approach extending the Theatre/ActorDEVS architecture in order to meet requirements of DEVS-World project. Provided extensions support architectural interoperability among heterogeneous DEVS simulators. The approach adopts previously described DEVS simulation protocol. At the moment, interoperability at modelling language level is not addressed. Each DEVS model is assumed to be implemented as a Java class complying with the ActorDEVS API (Cicirelli *et al.* 2008).

A *Coordinator* is introduced in order to coordinate the evolution of the overall simulation and it is in charge of implementing the DEVS simulation cycle (see Table 1). A *Configurator* makes it possible to configure the whole simulation system and start execution. An UML class diagram of system components is reported in Fig. 3.

The *Theatre* component and the *Configurator* are not exclusive of DEVS simulations, they are common to all actor-based applications. The *Coordinator* (see Fig. 4), instead, is tightly related to DEVS-World prospective.

A *DEVSControlMachine* has been purposely developed in order to work in pair with the coordinator and be compliant with the DEVS simulation protocol. This control machine implements a *CoupledSimulatorInterface*–like (see Fig. 2) and behaves as a DEVS simulator.

With respect to the approach proposed in (Xiaolin and Zeigler 2008) the *Coordinator* is only concerned with the execution of the DEVS simulation cycle. In particular it does not manage coupling information among DEVS models. Such information is directly handled at simulator level. In addition, being in a net-centric context, the *Coordinator* must wait until all outgoing messages, i.e. inter-simulator messages, are received by recipient simulators before proceeding to the applyDeltFunc phase (see Table 1). This is ensured by `Chek` messages (see Fig. 4) sent by simulators to the coordinator. Toward this, the `setCoordinator` method was added to *CoreSimulator* (see Fig. 3). `Chek` messages are actually generated at the end of sendMessages phase and after external messages are received.

CoupledSimulator interface (Fig. 3), which does not introduce further methods, extends both *CoreSimulator* and *Coupled* interfaces. This is to guarantee a clear separation of concerns among configuration (i.e. coupling management addressed by the *Coupled* interface) and simulation aspects (simulation protocol management addressed by the *CoreSimulator* interface).

It is worthy noting, finally, that a *DEVSControlMachine* is in charge of handling the simulation needs of all the models allocated to the same theatre. In other words, each theatre has one simulator instead of having one simulator for every distinct atomic model.

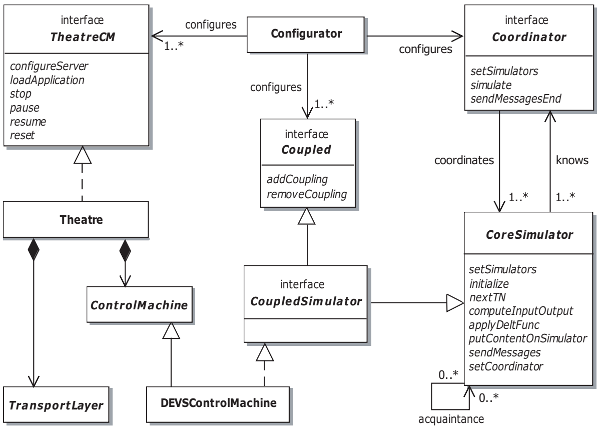


Figure 3: Class diagram of system components

```
public interface Coordinator{
   void setSimulators(SimulatorInfo[] si)throws Exception;
   void simulate(long simulationTime)throws Exception;
   void sendMessagesEnd(Check check)throws Exception;
}
```

Figure 4: Coordinator interface

In order to support the NCMS vision, a whole Theatre/ActorDEVS system, which can span from a single atomic model to a complex coupled model, is made usable through Web Services. Each system component is made available as a Web Service by means of specific objects called Wrappers. Client-side interactions are instead mediated by means of specific Proxy objects. It is worthy of note that in a service oriented architecture the roles of client and provider are not strictly defined, being possible for a same node to act as client or provider on the basis of the required/offered functionalities.

Wrappers and Proxies are transparently used. As a consequence, would e.g. Java RMI be used in place of Web-Services based protocols, only Wrappers and Proxies would be accordingly changed. Fig. 5 shows the architecture of a resultant Theatre/ActorDEVS system.

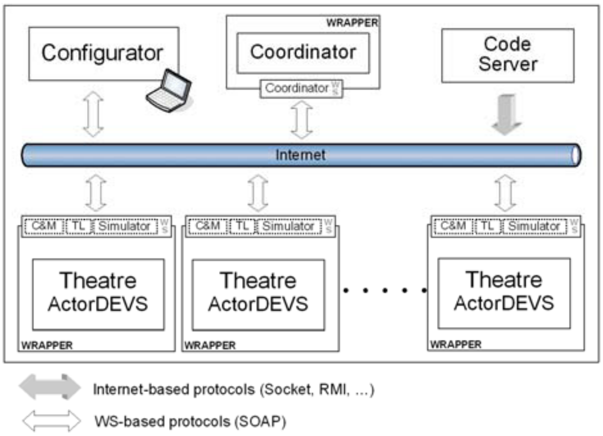


Figure 5: Architecture of a Theatre/ActorDEVS system

A *Code Server* is shared among theatres and it is used as a remote Java-class repository from which download the actor-based application to execute, i.e. in this case the DEVS models to simulate. Configuring and starting a simulation consists of four steps. The first step is devoted to setting-up the Theatre nodes by specifying the control machine, the transport layer to use and the code server IP address.

This is accomplished by exploiting the Configuration and Management Web Service (see the C&M-WS in Fig. 5). After the control machine is instantiated its functionality is made available as a Web Service which is automatically published (see the Simulator-WS in Fig. 5). The *DEVSControlMachine* oversees message exchange with other simulators. As a consequence, the transport layer (see the TL-WS in Fig. 5) in this scenario is used only to manage inter-theatre control messages.

The second step consists in assigning to each Theatre the DEVS model(s) to simulate. A single model may correspond to an atomic or to a coupled DEVS component. The Java class name of each model requires to be specified along with the parameters possibly required by its constructor. This step is carried out by exploiting the C&M-WS and completes when models get assigned to target theatres, i.e. downloaded from the code server and instantiated.

The third step consists in establishing the necessary bindings among coordinator and simulator services (i.e. acquaintance relationships). In particular, a *CoordinatorInfo* object is provided to each simulator and a list of all *SimulatorInfo* objects, relevant to simulators involved in the federation, is furnished to each simulator and to the coordinator. An info object contains the name of the service and the relevant service endpoint address which is necessary to contact and use it. As stated above, each simulator has to know the coordinator in order to communicate information about the state of the current sendMessages phase (see Table 1).

The fourth step consists in defining couplings among deployed models in order to build the entire simulation model. This is achieved by invoking the method *addCoupling* onto simulators. Coupling information mainly contains a couple of names, identifying the two ports to be connected. The first name is relevant to an output port of a DEVS component local to the simulator. The second name is relevant to an input port of a DEVS component which can be either local to the simulator or residing on a remote simulator. In the latter case, the name of the remote simulator is provided along with coupling information.

A naming policy is required to distinguish ports belonging to different instances of the same model. In particular, full name of a port is assumed to be specified in the form *modelInstanceName.portName*.

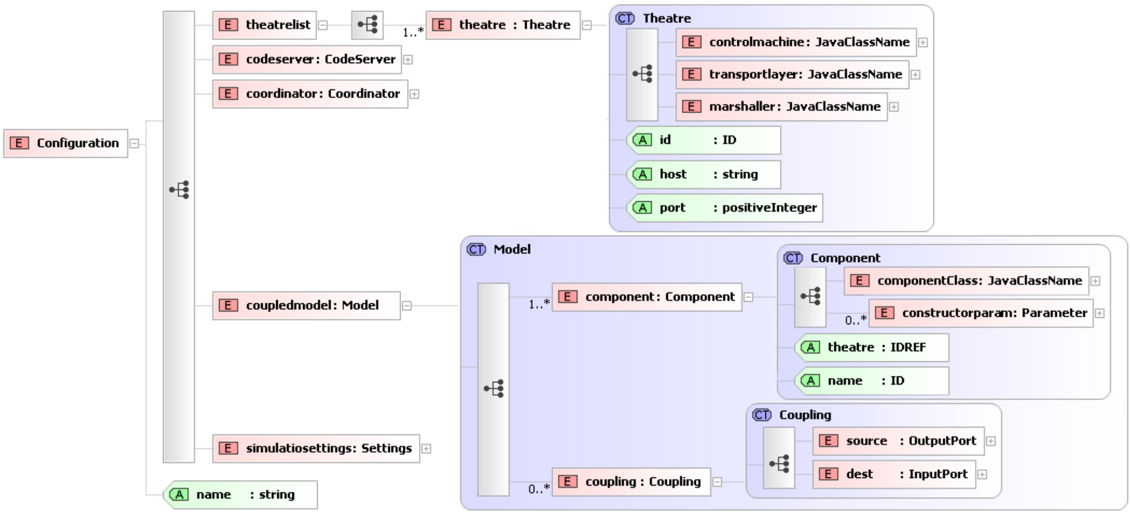


Figure 6: XML schema of the configuration files

At runtime, remote couplings get actualized by means of the so called *RelayPort* objects. Making a remote coupling implies linking an output port of a DEVS component to a relay port which, in turn, is logically connected to a remote input port. All of this makes the DEVS component unaware of network partitioning.

All data needed during configuration steps are contained in an XML file whose schema is reported in Fig. 6. In the current prototype system implementation, the Settings type is used only to contain the simulation time info. The CodeServer and Coordinator types contain information required to contact the relevant components on the web (e.g. service name, host, port). Other types are self-explanatory.

At configuration end, the Configurator may launch the simulation by calling the simulate method on the Coordinator which in turn triggers into execution the simulation control loop.

4. VARIABLE STRUCTURE SYSTEM EXAMPLE

The achieved implementation of WS-based Theatre/ActorDEVS architecture was tested by modelling and simulation of a variable structure system based on server relocation (Cicirelli *et al.* 2008). The modelled system consists of a collection (closed pipeline) of interconnected node components (see Fig. 7).

Each node receives from its environment a stream of jobs, stores them in a buffer (of unbounded size) and ultimately processes them using a number of server components. A system is assumed to work with a fixed number of servers. Servers cannot be dynamically generated because they model physical computing resources. However, a high loaded node can ask for a server to its neighbours. A dispatcher component in a node is in charge of handling the server relocation issues. Main difference between the model as handled in (Cicirelli *et al.* 2008) and here, consists in the achievement of structure dynamism.

In (Cicirelli *et al.* 2008), server components migrate from a node to another as mobile agents. In the scenario of this paper, though, servers do not migrate but port objects are created/destroyed dynamically in order to contact servers.

Asking for a server may return a server port through which a dispatcher can submit a job to a server allocated on a different node. As a consequence, server relocation is achieved by changing the number of servers a node can contact to process its jobs. Different strategies of server relocation can be considered (see later).

Fig. 7 depicts a three node system, together with input/output ports and connectors. Each node can direct useful statistical data to an external *Statistics* (*transducer*) component connected to the *StatOut* output port. When used, the *OverloadGenerator* can inject jobs randomly to any node.

Fig. 8 shows the internal structure of a node. Inter-node ports serve to send/receive an ask to/from a neighbour for a server (*ask-OUT?*, *askIN*), to send/receive a server to/from a neighbour (*moveIN?*, *moveOUT?*, *moveIN*), or to send/receive back a no longer useful server (*sendBackOUT?*, *sendBackIN*). Fig. 7 shows *delegate* connections (represented by using dashed lines) within a coupled node. The shadowed *TimerToken* component in Fig. 8 is required only by some relocation protocols.

A high loaded node, that is a node with a pending job but without idle servers, asks for a server port to its neighbours. When the *Dispatcher* of a node receives a request for a server, it honours the request with a server port if at least one idle server is available. Otherwise the request is ignored. If no server ports are obtained, a node asks again for a server port after a certain time delay. Three particular strategies (Cicirelli *et al.* 2008) were considered about the way a node can handle external utilizable servers.

On-demand strategy - A node which achieves an external server, views it as an own server. Therefore, the protocol freely distributes server ports among nodes on a on-demand basis. It can be anticipated that this strategy makes it possible for nodes to behave in a selfish-way, possibly leading to an unbalanced distribution of server use.

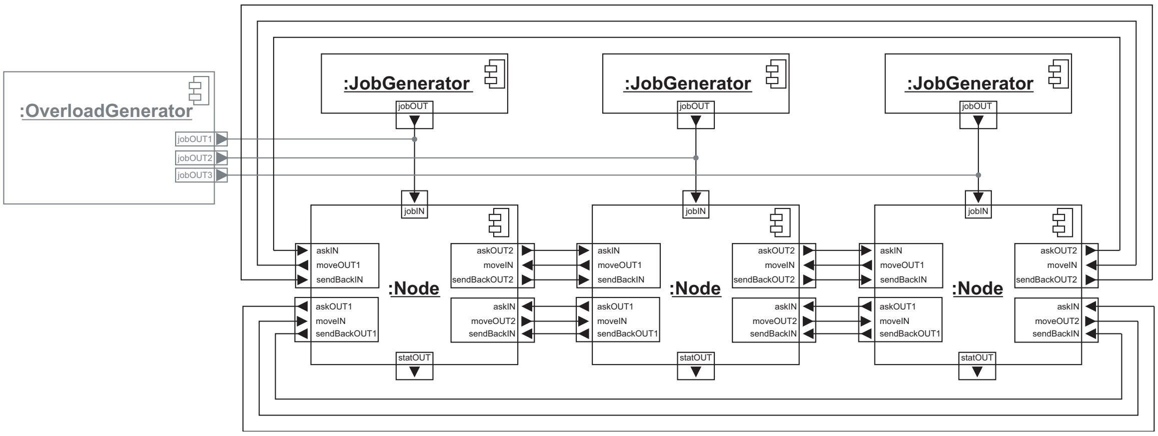


Figure 7: A ring of three nodes

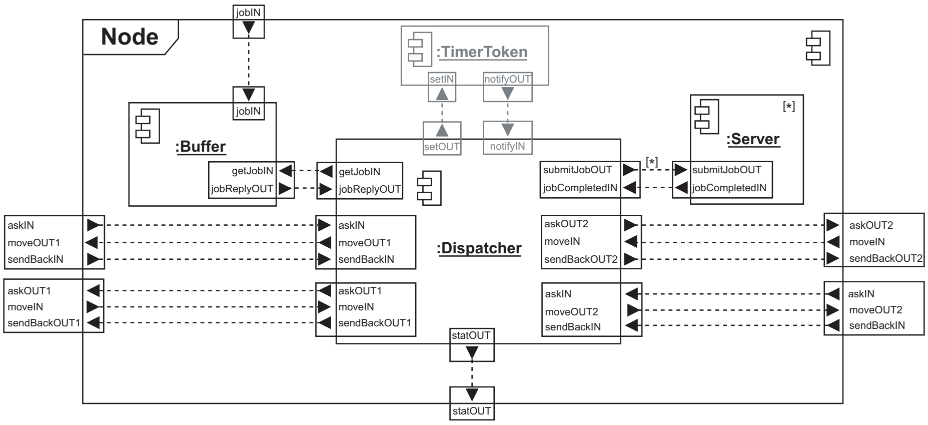


Figure 8: Internal structure of a node

Debt strategy - A debt concept for server allocation is introduced. A node which receives a server port from a neighbour, annotates the identification of the furnishing node. As soon as the *Dispatcher* of a debtor node has no pending job but has at least one idle server, it tries to exhaust its debits by anticipating restitution of some server ports to its creditor nodes. Intuitively, the protocol attempts to avoid non uniform utilization of servers.

Token passing strategy - One server port is used as a token which circulates upon the closed pipeline. A node receiving the token-server can use it if has a pending job but has no available local server. Otherwise, or after token usage, the token is forwarded to the next node in the ring. The strategy tries to anticipate a server request. A node which receives the token as well as server ports coming from neighbours, uses the token and sends back the other server ports.

5. CONFIGURATION, DEPLOYMENT AND SIMULATION

Some simulation experiments concerning the server relocation model described in the previous section were carried out by using two Theatre/ActorDEVS systems allocated on two Win platforms.

Another Win platform was used to host the Coordinator, the Code Server and the Configurator. The experiments were directed to study the effects of overloads starting from an equilibrium situation. Simulation parameters which, under either On-Demand or Debt strategy, ensure the buffers size or equivalently the mean delay time of jobs is definitely constant and of a low value are as follows.

The job interarrival time is in the interval [2,4], the job size (which indicates the time needed to process the job) belongs to the interval [8,15]. The time delay a *Node* waits between two consecutive asks for a server was set to 1 time unit. The number of servers initially allocated to each node is 4. Starting from the equilibrium, the *OverloadGenerator* (see Fig. 7) is capable of injecting each generated job to a randomly chosen node.

To respond to the overload, one additional server was introduced, whose management ultimately depends on the adopted strategy(ies).

For instance, under On-demand or Debt strategies the extra server is initially assigned to a given node. In the Token passing strategy, instead, the extra server (its port) circulates in the pipeline ring. In this case, to avoid Zeno behaviours, the token which reaches the node where it was last used, is forced to wait one time unit before starting the next round.

The job mean delay time (that is the time which elapses between the instant in time a job is received by Buffer and the subsequent time the job gets assigned to a server) was measured by the Statistics components. The investigated strategies for responding to overload were: Debt & Token, On-demand & Token, On-demand alone.

The DEVS models relevant to *Node*, *JobGenerator*, *OverloadGenerator* and *Statistics* were deployed to the Code Server. A number of *Node*s, varying from one to five, along with the relevant instances of *JobGenerator*s were assigned to each Theatre. The *OverloadGenerator* and the *Statistics* were allocated on a single Theatre. The simulation time limit was set to $t_{END}=10^5$.

Different system configurations were actualized by specifying different configuration files. An excerpt of such a file is reported in Fig. 9. The configuration is relevant to a relocation system model made up of two *Node*s allocated to two theatres. Only the Debt strategy is considered.

Coupling information, common to all the configuration files, is used to build up the overall simulation model. In particular:

- each *JobGenerator* was coupled with the relevant *Node*
- each *Node* was coupled with its neighbors in the closed pipeline
- the *OverloadGenerator* was coupled with all the *Node*s
- each *Node* was coupled with the *Statistics*.

```
<?xml version="1.0" encoding="utf-8"?>
<Configuration name="RelocationServers"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="./TheatreDEVS.xsd">
<theatrelist>
 <theatre id="PERSEUS8000" host="perseus" port="8000">
  <controlmachine name="theatre.DEVSControlMachine"/>
  <transportlayer name="theatre.transport.WSTransport"/>
  <marshaller name="theatre.marshaler.ByteArrayMrshlr"/>
 </theatre>
 <theatre id="HYDRA8000" host="hydra" port="8000">
  ...
 </theatre>
</theatrelist>
<codeserver url="http://orion:8989"/>
<coordinator name="Coordinator" host="orion" port="8080"/>
<coupledmodel>
 <component name="Node1" theatre="PERSEUS8000">
  <componentClass name="relocation.Node"/>
  <!-- number of servers -->
  <constructorparam type="java.lang.Long" value="4"/>
  <!-- token disabled -->
  <constructorparam type="java.lang.Boolean" value="false"/>
  <!-- debt enabled -->
  <constructorparam type="java.lang.Boolean" value="true" />
 </component>
 <component name="Node2" theatre="HYDRA8000">
  <componentClass name="relocation.Node"/>
  ...
 </component>
 <component name="OverloadGenerator" theatre="PERSEUS8000">
  <componentClass name="relocation.OverloadGenerator" />
 </component>
 ...
 <coupling>
  <source theatre="PERSEUS8000" port="Node1.sendBackOut2"/>
  <dest theatre="HYDRA8000" port="Node2.sendBackIn1"/>
 </coupling>
 <coupling>
  <source theatre="PERSEUS8000" port="Node1.askOut2"/>
  <dest theatre="HYDRA8000" port="Node2.askIn"/>
 </coupling>
 <coupling>
 ...
</coupledmodel>
<simulatiosettings>
  <simulationtime>100000</simulationtime>
</simulatiosettings>
</Configuration>
```

Figure 9: An excerpt of a configuration file

Coupling information dictates system topology at configuration time. At runtime, on the basis of the adopted strategy, a *Node* may dynamically change the servers it actually contacts without resorting to the add/remove coupling mechanism.

Simulation experiments (see Fig. 10) indicate that the combination of Debt & Token strategies minimizes the job mean delay time when compared to the other strategies.

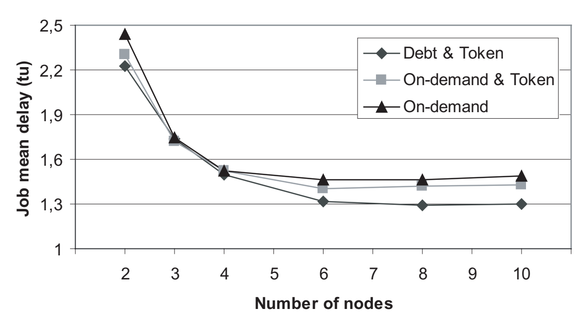


Figure 10: Job mean delay time vs. number of nodes

6. CONCLUSIONS

A prototype version of the Theatre/ActorDEVS architecture based on Web Services was realized and tested. The implementation relies on Java technology. In particular, the SOAP engine Axis (Axis website) is used for managing WS related aspects. The following are some points which deserve some discussion within the community of DEVS-World.

- The DEVS simulator protocol appears “too much synchronous” for a networked context. Many interactions among the simulation-protocol participants are required for each simulation step independently from the complexity of the simulated model. A systematic exploitation of a kind of “lookahead” could alleviate the problem. By exploiting lookahead the coordinator could give a granted time to each simulator allowing a more independent evolution of local simulation.
- Another (obvious) issue concerns simulation performance achievable by the use of WSs. This is not only tied to the use of verbose XML for SOAP messaging but mainly to the management of network connections. Simulation experiments confirmed that network resources (connections) of operating system may be wasted considerably during simulation and need in general careful control.

On-going work is directed at:

- improving the Configurator component by providing a friendly GUI for visual system configuration, model composition, deployment and simulation control
- replacing Axis by other Web Service infrastructure e.g. related to latest J2EE
- introducing a model repository service, enabling model reuse and sharing
- adopting standard formalisms like DEVSML for supporting DEVS modelling
- favouring model and experiments interchange by developing translation tools allowing model transformation from a high-level implementation-independent formulation into the terms of a specific DEVS setting (e.g. ActorDEVS and Java) and vice versa
- experimenting with Theatre/ActorDEVS architecture in an heterogeneous environment where diverse DEVS simulators have to cooperate
- developing tools for visual modelling.

REFERENCES

Agha, G., 1986. *Actors: A model for concurrent computation in distributed systems*. Cambridge, MIT Press.

Axis website. Available from: http://ws.apache.org/axis/index.html. [Accessed May 2008].

Cicirelli, F., Furfaro, A., Giordano, A., Nigro, L., 2007a. An agent infrastructure for distributed simulations over HLA and a case study using Unmanned Aerial Vehicles. *Proceedings of 40th Annual Simulation Symposium*, IEEE Computer Society Press, pp. 231-238, March, Norfolk (VA).

Cicirelli, F., Furfaro, A., and Nigro, L., 2006. A DEVS M&S framework based on Java and actors. *Proceedings of 2nd European Modelling and Simulation Symposium (EMSS 2006)*, pp. 337-342.

Cicirelli, F., Furfaro, A., and Nigro, L., 2007b. Conflict management in PDEVS: an experience in modelling and simulation of time Petri nets. *Proceedings of Summer Computer Simulation Conference (SCSC'07)*, pp. 349-356.

Cicirelli, F., Furfaro, A., and Nigro, L., 2007c. Integration and interoperability between Jini services and Web Services. *Proceedings of IEEE Int. Conf. on Services Computing (SCC'07)*, pp. 278-285, July.

Cicirelli, F., Furfaro, A., and Nigro, L., 2008. Actor-based Simulation of PDEVS Systems over HLA. *Proceedings of 41st Annual Simulation Symposium (ANSS'08),* pp. 229-236, April, Ottawa, Canada.

Cicirelli, F., Furfaro, A., Nigro, L., and Pupo, F., 2007d. A component-based architecture for modelling and simulation of adaptive complex systems. *Proceedings of 21st European Conference on Modelling and Simulation (ECMS'07d)*, 4-6 June, Prague.

DEVS World, 2007. DEVS_WORLD: A platform for developing advanced discrete-event simulation at worldwide scale. Internal document.

Hu, X., Zeigler, B.P., and Mittal, S., 2005. Variable structure in DEVS component-based modelling and simulation. *Simulation*, 81(2), 91-102.

Hu, X., and Zeigler, B.P, 2004. Model continuity to support software development for distributed robotic systems: A team formation example. *J. of Intelligent and Robotic Systems*, 39(1), 71-87.

Mittal, S., Zeigler, B.P., Martin, J.L.R., Sahin, F., and Jamshidi, M., 2008. Modeling and simulation for systems of systems engineering. In: *System of Systems – Innovations for the 21st Century*, Wiley (in press).

Papazoglou, M.P., and Georgakopulos, D., 2003. Service Oriented Computing. *Communications of the ACM*, 46(10), 25-28.

Xiaolin, H., and Zeigler, B.P., 2008. A Proposed DEVS Standard: Model and Simulator Interfaces, Simulator Protocol, Internal document.

Yu, Y.H., and Wainer, G., 2007. eCD++: an engine for executing DEVS models in embedded platforms. *Proceedings of SCS Summer Simulation Multiconference*, pp. 323-330.

Zeigler, B.P., Praehofer, H., and Kim, T.G., 2000. *Theory of modeling and simulation*. 2nd edition, New York, NY, Academic Press.

Wooldridge, M., 2002. *An introduction to multi-agent systems*. John Wiley & Sons, Ltd.

TEMPORAL ANALYSIS OF COMPLEX TIME-DEPENDENT SYSTEMS: AN APPROACH BASED ON TIME PETRI NETS, ACTORDEVS AND HLA

Franco Cicirelli[a], Angelo Furfaro[b], Libero Nigro[c], Francesco Pupo[d]

[a] [b] [c] [d] Laboratorio di Ingegneria del Software
(www.lis.deis.unical.it)
Dipartimento di Elettronica Informatica e Sistemistica
Università della Calabria
87036 Rende (CS) – Italy

[a]f.cicirelli@deis.unical.it, [b]a.furfaro@deis.unical.it, [c] l.nigro@unical.it, [d]f.pupo@unical.it

ABSTRACT

The design of time-dependent systems is challenging because it must fulfil both functional and temporal requirements. A properly abstracted model of one such a system, with temporal aspects only, is often derived and analyzed in order to evaluate the temporal behaviour of the system. Temporal analysis can be based on simulation or (hopefully) on exhaustive state space exploration. The latter techniques, though, are difficult to practice for large system models. In the work described in this paper, Time Petri Nets (TPNs) are preferred to formalize a time-dependent system because they facilitate the expression of concurrency, distribution, synchronization, mutual exclusion etc. concerns. An approach is proposed where a TPN model is mapped on ActorDEVS, a minimal and efficient Java framework supporting parallel or interleaved DEVS model execution. Complex TPN models can be analyzed using distributed simulation of ActorDEVS over HLA. The approach is demonstrated by means of a real-time realistic example.

Keywords: DEVS modelling and simulation, time Petri nets, actors, HLA

1. INTRODUCTION

The design of systems with timing constraints (e.g. embedded real-time systems, communication protocols, flexible manufacturing systems etc.) is difficult because it must fulfil both functional and (most importantly) temporal requirements. A properly abstracted model of one such a system, where only temporal aspects are explicitly modelled, is often derived and analyzed in order to evaluate the temporal behaviour of the system.

Temporal analysis can be based on simulation or (hopefully) on exhaustive state space exploration using e.g. model checking techniques (Cicirelli et al., 2007c). The latter techniques, though, are difficult to practice in the case of large system models which can have a large or even unbounded state graph.

This paper proposes an approach to modelling and temporal analysis of embedded control systems (Furfaro and Nigro, 2007) which is based on Time Petri Nets (TPNs) (Merlin and Farber, 1976; Cicirelli *et al.*, 2007c) and simulation. The approach is novel and maps preliminarily a TPN model on to ActorDEVS (Cicirelli *et al.*, 2008), a lean and efficient agent-based framework in Java supporting Parallel DEVS (Zeigler *et al.*, 2000). A distinguishing feature of ActorDEVS with respect to standard DEVS tools like DEVSJAVA (Zeigler and Sarjoughian, 2003) concerns the possibility of customizing the simulation engine in order to cope with different execution semantics. As a significant example, an interleaved parallel simulation engine was developed which is able to manage at runtime conflicts existing among TPN transitions (Cicirelli *et al.*, 2007b). The realization is beyond the scope of standard DEVS because the built-in semantics of maximal parallelism assumed by conventional simulation infrastructure (Zeigler and Sarjoughian, 2003), implies that all components which can undergo a simultaneous state transition at a given time, must do so and then cannot take care of conflicting situations.

Application of the proposed approach proceeds as follows. First a TPN model is visually designed and modularized in the context of the TPN Designer toolbox (Carullo *et al.*, 2003; Cicirelli *et al.*, 2007c). Then the model is translated into PNML (Billington *et al.*, 2003). The PNML version, finally, is partitioned, deployed and executed over a certain number of computing nodes, using the runtime support for TPNs achieved with ActorDEVS and the services of Theatre/HLA (Cicirelli *et al.*, 2008) for distributed simulation.

The paper demonstrates the use of the approach by modelling and analysis of a realistic real-time system related to a traffic light controller which is capable of responding to the exceptional situation corresponding to the arrival of an ambulance which must be handled within required safety and timing constraints.

2. BASIC CONCEPTS OF TIME PETRI NETS

A TPN is assumed to be a tuple $TPN = (P, T, A, I_{nh}, W, M_0, I^S)$ where

- P and T are non empty and disjoint sets respectively of places and transitions of the underlying Petri net (Murata, 1989)

- A is a set of arcs: $A \subseteq P \times T \cup T \times P$
- I_{nh} is a set of inhibitor arcs: $I_{nh} \subseteq P \times T$
- W associates weights to arcs: $W : A \cup I_{nh} \rightarrow \mathbf{N}$, with $\mathbf{N}$ the set of natural numbers. Weights are assumed strictly positive for arcs in A, 0 for inhibitor arcs
- M_0 is the initial marking: $M_0 : P \rightarrow \mathbf{N}$ in the usual sense of Petri nets
- I^S is the static firing interval function: $I^S : T \rightarrow \mathbf{R} \times (\mathbf{R} \cup \{\infty\})$.

Place $p \in P$ is an *input place* for transition t if there is an arc (p,t) in A. Place p is an *inhibitor place* for t if $(p,t) \in I_{nh}$, i.e. there exists an *inhibitor arc* connecting p to t. An inhibitor arc is graphically represented by a dot terminated line. A place p is an *output place* for t if there exists an arc (t,p) in A. The set of input and inhibitor places of t is said its *preset* and denoted by ${}^\bullet t$. The set of output places constitutes the transition *postset* which is denoted by $t^\bullet$.

I^S associates with each transition t a *dense* firing interval whose bounds are assumed to be specified by non negative reals: $I^S(t) = [a,b]$ with $0 \le a \le b$, b can be ∞. Bound a is said the (static) *earliest firing time* (EFT^s) of t, b the (static) *latest firing time* of t (LFT^s).

Let M be a marking. Transition t is said enabled in M, denoted by $M[t>$, iff

$$M[t> \Leftrightarrow \forall p \in {}^\bullet t \begin{cases} M(p) \ge W(p,t) & \textit{if } (p,t) \notin I_{nh} \\ M(p) = 0 & \textit{if } (p,t) \in I_{nh} \end{cases}$$

As soon as a transition t is enabled, it *starts* firing (server semantics). The firing end event is constrained to occur in the time interval associated with the transition. Let τ be an instant in time when transition t is enabled. Provided t is continuously enabled, t cannot fire before $\tau + a$ but *must* fire before or at $\tau + b$, unless it is disabled by the firing of another transition. At the time transition firing ends, tokens are removed from the input places and new tokens are generated in to output places as in classic Petri nets. Let M_{before} be the net marking just before t completes its firing. Firing end of t transforms M_{before} in M_{after}, denoted by $M_{before}[t > M_{after}$, by an instantaneous and atomic process in two phases:

(phase 1-token withdrawal)

$\forall p \in P$ ***if*** $p \in {}^\bullet t$ ***then*** $M'(p) = M_{before}(p) - W(p,t)$

else $M'(p) = M_{before}(p)$ ***endif***

(phase 2-token deposit)

$\forall p \in P$ ***if*** $p \in t^\bullet$ ***then*** $M_{after}(p) = M'(p) + W(t,p)$

else $M_{after}(p) = M'(p)$ ***endif***

where M' represents the intermediate marking generated after token withdrawal. It is worth noting that an enabled transition t', *i.e.* $M_{before}[t \wedge M_{before}[t'$, can be disabled (its firing stopped) by the firing of t, either in marking M' (because of a conflict due to the sharing of some input places with t, i.e. ${}^\bullet t \cap {}^\bullet t' \neq \varnothing$, or in the reached marking M_{after} (because of the existence of some inhibitor arc: $\exists p \in t^\bullet : (p,t') \in I_{nh}$). Similarly, a disabled transition t'', i.e. $M_{before}[t \wedge \neg M_{before}[t''$, due to the firing of t can become enabled in M' or in M_{after}. Single server firing semantics is assumed. After its own firing, would t be still enabled it is considered as any new enabled transition.

3. A TRAFFIC LIGHT CONTROLLER

In order to illustrate the approach, modelling and simulation of a Traffic Light Control system (TLC) (Raju and Shaw, 1994) are described in the following. In the proposed scenario, the traffic flow at an intersection between an avenue and a street is regulated by two traffic lights. The lights are operated by a control device (controller) that, in normal conditions, alternates in a periodic way the traffic flow in the two directions. In addition, the controller is able to detect the arrival of an ambulance and to handle this exceptional situation by allowing the ambulance crossing as soon as possible and in a safe way. For the sake of simplicity, it is assumed that at most one ambulance can be in the closeness of the intersection at a given time.

During normal operation conditions, the sequence green-yellow-red is alternated on the two directions with the light held green for 45 seconds, yellow for 5 seconds and red on both directions for 1 second. The intersection is equipped with sensors able to detect the presence of an ambulance at three different positions during its crossing. As soon as the ambulance arrival is detected, a signal named "approaching" is sent to the controller. When the ambulance reaches the nearness of the intersection the signal "before" is issued. After the ambulance completes the crossing the signal "after" is generated. The controller reacts to the "approaching" event by leading the intersection to a safe state, i.e. bringing both lights on red.

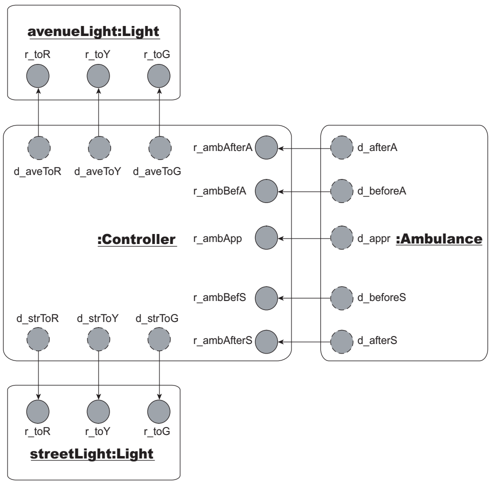


Figure 1: TLC system model

When the signal "before" is received, the controller switches to green the light on the ambulance's arrival direction. After the ambulance leaves the intersection ("after" event) the controller turns the green light to red and resumes its normal sequence.

Fig. 1 illustrates a model of the TLC system which is made of four connected components: there are two instances of the Light component, which respectively correspond to the light on the avenue and that on the street, one Ambulance component, which models the behaviour of the sensing equipments of the intersection and one Controller component, which implements the above described control logic.

Each component is specified by exploiting the *module* construct available in PNML (Billington *et al.*, 2003). Modules support the creation of several independent instances of a given sub-net, usable in different contexts. For example, in Fig. 1, aveLight and streetLight are two instances of the module Light. Module interfaces are defined by means of *import* and *export* places. An export place (represented by gray-shaded disc with continuous border) is a place that is made visible outside the module; an import place (represented by gray-shaded disc with dashed border) is a reference to a place owned by another module instance. Connections among module instances are achieved, as in Fig. 1, by linking each import place to an export place.

The behaviour of a module is modelled by a TPN sub net. Fig. 2 details the TPN model of the module Light. The places red, yellow and green model the status of the traffic light. Places r_toR, r_toY, and r_ToG are the export places of the component interface.

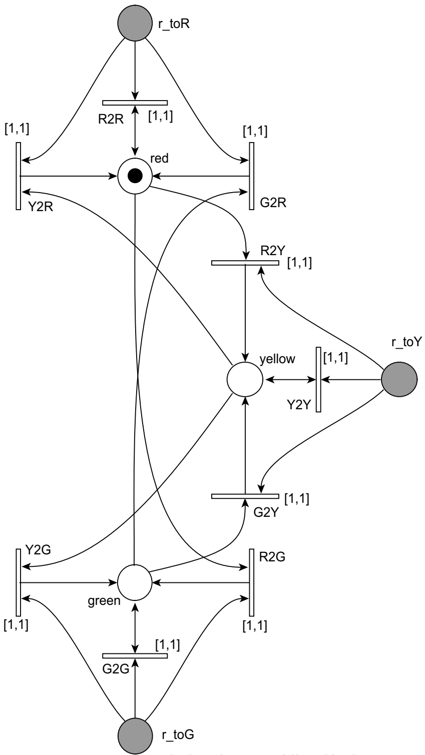


Figure 2: TPN model of a traffic-light.

An external component may ask to switch the light on green, yellow or red by respectively putting a token in the place r_toG, r_toY or r_toR. The initial marking of this sub net, where only place red contains a single token, models the fact that at start-up the red light is on and the others are off.

As can be seen from the time-windows of the transitions if Fig. 2, the handling of each request to change the status of a light, requires 1 time unit to be served.

Fig. 3 depicts the TPN model of the Ambulance module, which is used to simulate the sporadic arrival of ambulances needing to cross the intersection. This module has an interface made of five import places, which are used to notify the ambulance movements. It can be easily noticed that this net is symmetrical, with the upper part which models the crossing along the avenue and the lower part that along the street. The initial marking corresponds to a situation where the ambulance chooses its next crossing direction in a non deterministic way. This is modelled by the conflict existing between the transitions NextS and NextA.

The ambulance approaching on the avenue (street) is signalled by the firing of transition ApprA (ApprS) that puts a token into the import place d_appr. Timing specifications of these two transitions corresponds to the minimal and maximal interval between two successive ambulance arrival events. Transition BefA (BefS) puts a token into the import place d_beforeA signalling that the ambulance finds itself just before the intersection and that it is coming from the avenue (street). Timing constraints of this transition correspond to the time interval that may elapse between an approaching and a before event. Firing of transition AfterA (AfterS) corresponds to the completion of the ambulance crossing along the avenue (street) and results in a token put in the import place d_afterA (d_afterS).

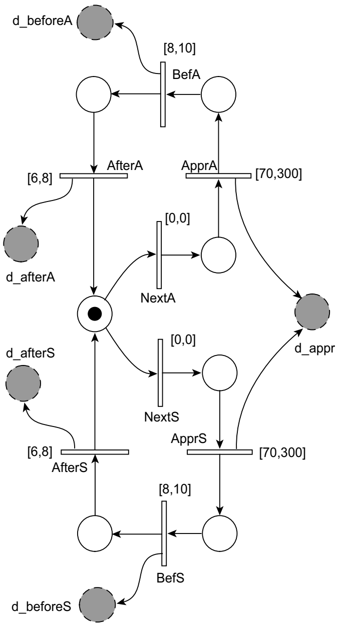


Figure 3: TPN model of the ambulance

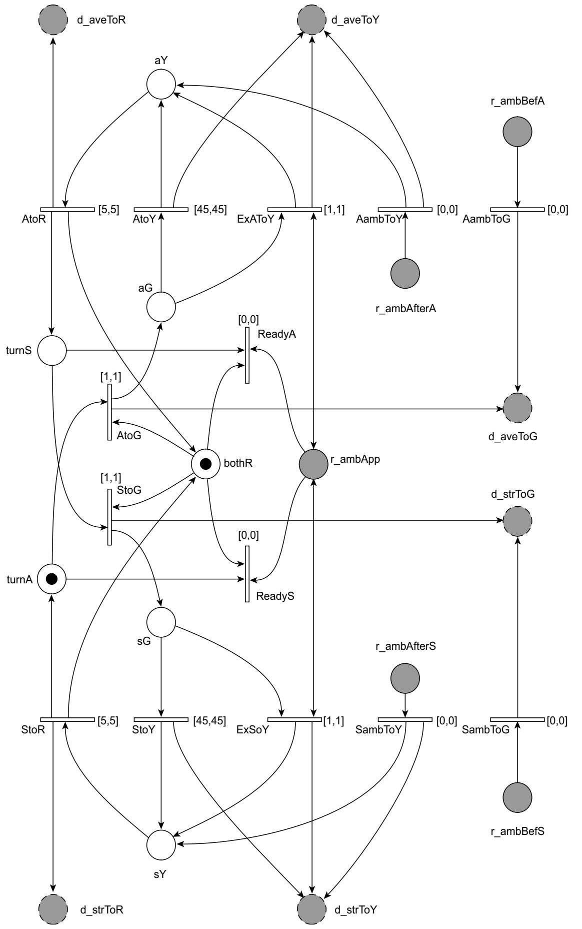


Figure 4: TPN model of the controller

The time interval associated to AfterA (AfterS) models the time needed to the ambulance to complete its crossing.

Fig. 4 is the TPN model for the controller which is the most complex component of the system. Also this net is symmetrical with the upper part interacting with the avenue light and the lower part with the street light. At start-up, the controller assumes that both lights are red and this is reflected by the presence of a token in the place bothR. The marking of places turnA and turnS determines whether the sequence green-yellow-red must starts respectively on the avenue or on the street. Under a normal operation mode, if there is a token into turnA, transition AtoG fires after 1 sec, puts one token into aG and another one into the export place d_aveG asking the light in the avenue to switch on green. After 45 sec, transition AtoY fires and puts one token into aY and another one into d_aveY, then after 5 sec transition AtoR fires and generates one token into places bothR, turnS, and d_aveR. Place turnS is now marked and, after 1 sec, the sequence starts again but on the street. This cycle continues until the approaching of an ambulance is notified by a token arrival into the export place r_ambAppr. The controller reaction to this event depends on the current status of the lights. It must bring both lights to red in a safe way in order to be ready to handle the ambulance crossing. The best case occurs when both light are already red and then one of the two transitions readyS or readyT can fire and disable the beginning of another sequence. In the case the light on the avenue (street) is green, the presence of one token into r_ambAppr and of one another into aG (sG) enables the immediate transition ExAtoY (ExStoY) whose firing has the same effect as the firing of aY (sY), i.e. asking the avenue (street) light to switch on yellow. After that, the sequence continues as in the normal case until both lights become red and the start of the next sequence is avoided as before. No special provision has to be taken when one of the two light is yellow. After handling the ambulance approaching, the controller maintains both lights red until the event of a token arrival into place r_ambBefA (r_ambBefS) notifies that the ambulance is just before crossing the intersection along the avenue (street). The controller reacts switching to green the light on the avenue (street) by firing transition AambToG (SambToG). The light is maintained green until the ambulance completes the crossing, event that is notified by a token into the import place r_ambAfterA (r_ambAfterB). This token enables the immediate transition AambToY (SambToY) whose firing has the same effect as that of AtoY (StoY), i.e. asking the light to switch on yellow. After 5 sec transition AtoR (StoR) fires and regenerates one token into place turnS (turnA). At this point, the controller restarts its normal operation mode.

4. TLC PROPERTY ANALYSIS

The behaviour of the TLC system can be validated by simulating its TPN model. System validation rests on checking that a set of assertions about its logical and temporal behaviour are satisfied at certain points during simulation, i.e. when events of interest occurs.

An example of safety property, i.e. one that must always be satisfied during system evolution, regards the consistent status of the traffic lights.

In order to avoid accidents among vehicles crossing the intersection, when on a direction the light is green or yellow, thus allowing the traffic on this direction, the light on the opposite direction must be red. This property can be checked by inspecting the marking of the two instances of the Light component each time one of their transitions fires. When such an event occurs, if one between avenueLight.green and avenueLigth.yellow is marked then streetLight.red must be marked and if one between streetLight.green and streenLigth.yellow is marked then avenueLight must be marked.

Another safety property concerns the status of the intersection at the time a “before” event is received. When such an event occurs no vehicle should be allowed to cross the intersection, i.e. the lights should be red on both directions. This property can be checked by inspecting the marking of places avenueLight.red and streetLight.red when one between transitions BefA and BefS fires in the Ambulance component.

Assuming that it takes at least 4 sec for the ambulance to reach the intersection from the time instant of the before signal, it follows that there exists a deadline of 3 sec for turning green the light on the arriving direction. This accounts for the fact that a light takes 1 sec for changing its status. This property can be checked by recording the occurrence time of last firing of transition BefA (BefS) and measuring the elapsed time when the corresponding light is turned green, i.e. when one of the transitions having avenueLight.green (streetlight.green) in its postset fires.

The correct sequencing of the lights on each direction can also be easily checked by listening transition firing of Light components. A correct behaviour requires that only transitions RtoG, GtoY, and YtoR may fire: the firing of a transition out of this set denotes a wrong sequence.

5. CONCEPTS OF ACTORDEVS OVER THEATRE/HLA

A modular TPN model can be partitioned and deployed for execution over an instance of the Theatre architecture (Cicirelli *et al*., 2007a). ActorDEVS (Cicirelli *et al*., 2007b-2008) supports Parallel DEVS component development "in-the-small". Theatre furnishes the mechanisms required for distributed simulation, e.g. built on top of HLA middleware (DMSO, on-line; Kuhl *et al*., 2000) which provides time management and communication services to the theatres (federates) which compose the whole system (federation). For each ordered pair of communicating theatres a corresponding *interaction class* (Cicirelli *et al*., 2007a) is introduced and used as a communication channel.

ActorDEVS is agent-based. Actors with asynchronous message-passing are the basic building blocks, supporting DEVS atomic/coupled models. Message processing is atomic and cannot be pre-empted. In the mapping of TPN onto ActorDEVS, transitions correspond to atomic components. Places are topological entities which, as soon as they change their internal markings, alert dependent transitions to check their enabling status. The sub net assigned to a given theatre is a flattened coupled component. Flattening allows the use of one simulator per theatre, thus the simulation infrastructure of standard DEVS which associates a simulator to each atomic or coupled model (a multi-threaded organization) is avoided. ActorDEVS design minimizes the number of exchanged messages during simulation and then favours high-performance execution.

Fig. 5 is a snapshot of a typical Theatre system over HLA. A fundamental component in a theatre is the ControlMachine which is responsible of local message scheduling and dispatching. Local actors are held within the Local Actor Table (LAT). Since ActorDEVS actors can migrate between theatres, a Network Class Loader (NCL) is in charge to retrieving "on-the-fly" the class of a foreign received object from a network repository and loading it in the local JVM.

Specific control machines for ActorDEVS, working with HLA under conservative distributed simulation, were developed. They include ParallelSimulationEngine and InterleavedSimulationEngine.

ParallelSimulationEngine, detailed in (Cicirelli *et al*., 2008), follows standard Parallel DEVS execution semantics. InterleavedSimulationEngine, on the other hand, fires one component (e.g. one transition of a TPN sub model) at a time. This is a key for proper management of conflicts (Cicirelli *et al*., 2007b), when the firing of a transition can disable transitions which share some input places. Other possibilities for disabling are related to the use of inhibitor arcs (see also next section).

ParallelSimulationEngine uses a combination of virtual (simulation) and logical times in order to ensure causality relationships among simultaneous events, when concurrent and interacting components are allocated to different physical nodes, are ultimately fulfilled.

6. TPN MODEL PARTITIONING

The enabling of a transition depends on the marking of the places of its preset. In this work, network message exchanges are purposely avoided during the enabling process. For this reason a model partitioning is assumed where a transition and the places of its preset are deployed on the same node (theatre). This implies that two transitions whose presets share at least one place (structural conflict) must also find themselves on the same node. This fact restricts the number of ways a model can be partitioned. By tracking this type of dependency among the transitions of a TPN model it is possible to determine the maximal number of partitions that can be obtained. These basic partitions are defined as equivalence classes induced by an equivalence relation.

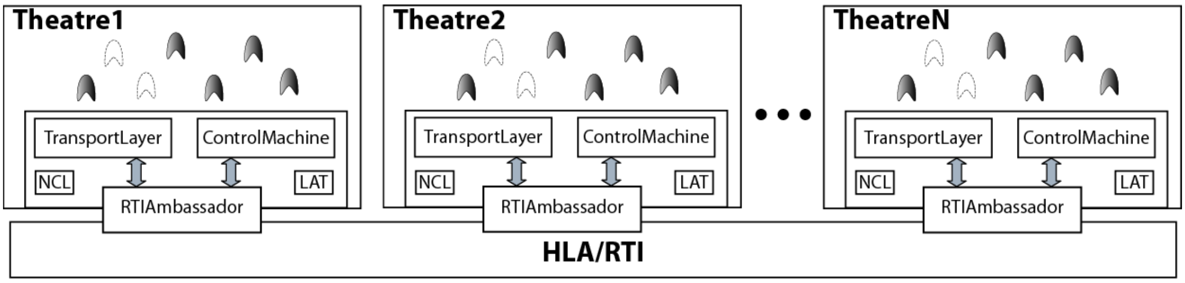


Figure 5: A Theatre federation over HLA

Let R be a binary relation among the transitions of a TPN defined as $R = \{(t_1, t_2) \in T \times T : {}^\bullet t_1 \cap {}^\bullet t_2 \neq \varnothing\}$, where $\varnothing$ denotes the empty set. R is reflexive and symmetrical. Let R^* be the transitive closure of R. R^* is an equivalence relation and as such it naturally induces a partitioning of T given by the quotient set T / R. The equivalence classes induced by R^* constitute the basic partitions of the model. Transitions belonging to the same equivalence class must be allocated on the same node. A model where R^* induces a unique equivalence class cannot obviously be partitioned.

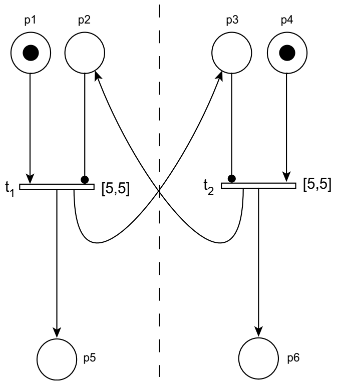


Figure 6: A conflict situation due to inhibitor arcs

The preceding concepts are sufficient to handle TPN models without inhibitor arcs. However, when inhibitor arcs are used, another type of dependency among transitions must be taken into account (see also the example in Fig. 6).

Transitions t_1 and t_2 of Fig. 6, would have been deployed on two different nodes because they are part of two distinct equivalence classes of R^*. However, between these transitions there exists a conflict situation due to the presence of inhibitor arcs. The firing of t_1 disables t_2 by putting a token into place p_3. For an analogous reason the firing of t_2 disables t_1.

To account for these type of conflicts, a coarser equivalence relation has to be used. Let RH be a binary relation defined as:

$$RH = \{(t_1, t_2) \in T \times T : ((t_1, t_2) \in R \vee \exists p \in t_1^\bullet : (p, t_2) \in I_{nh})\}$$

RH takes care of dependencies induced by inhibitor arcs and the equivalence classes induced by RH^*, its reflexive-symmetrical-transitive closure, can be used to establish basic partitions.

7. DEPLOYMENT AND SIMULATION

The TLC TPN model was partitioned according to the techniques described in the previous section and actually deployed as a federation of two theatres over HLA. One theatre hosts the traffic lights and the ambulance components, the other hosts the controller. Listing 1 shows a fragment of PNML describing the module for street lights.

```
<pnml>
  <module name="Light">
    <interface>
      <exportPlace ref="toR" id="r_toR"/>
      <exportPlace ref="toY" id="r_toY"/>
      <exportPlace ref="toG" id="r_toG"/>
    </interface>
    <!-- places -->
    <place id="toR"/>
    <place id="red">
      <initialMarking><text>1</text></initialMarking>
    </place>
    <place id="toY"/>
    <place id="yellow"/>
   ...
    <!-- transitions -->
    <transition id="R2R">
      <firetime>
        <lBound>1</lBound>
        <uBound>1</uBound>
      </firetime>
    </transition>
    <transition id="R2G">
        <firetime>
          <lBound>1</lBound>
          <uBound>1</uBound>
        </firetime>
    </transition>
    ...
   <!-- arcs -->
    <arc source="toR" target="R2R">
      <weight><text>1</text></weight>
      <type direction="PT"/>
    </arc>
    <arc source="toR" target="R2G">
       <weight><text>1</text></weight>
       <type direction="PT"/>
    </arc>
    <arc source="toR" target="R2Y">
       <weight><text>1</text></weight>
        <type direction="PT"/>
    </arc>
    <arc source="red" target="R2R">
        <weight><text>1</text></weight>
        <type direction="PT"/>
    </arc>
    <arc source="R2R" target="red">
        <weight><text>1</text></weight>
        <type direction="TP"/>
    </arc>
    ...
  </module>
</pnml>
```

Listing 1: A portion of PNML module Light

Listing 2 illustrates the content of a PNML file which describes how the TLC model is built by instantiating and connecting the modules of the various components and how it should be partitioned and deployed in order to be simulated. Model partitioning is achieved by exploiting the PNML page construct. Each page contains a part of the model that constitutes a unit of deployment. One or more pages can be deployed for execution on a given physical node. In this example there are two pages respectively named left and right.

The first page contains two instances of the module Light and one instance of the module Ambulance. The second page contains only one single instance of the Controller. Each page also describes how the various instances are interconnected, i.e. the bindings among reference places and actual places.

```
<?xml version="1.0" encoding="utf-8"?>
<pnml>
 <net id="TLC_Prova" type="TPNNet">
  <toolspecific>
   <lp ip="HYDRA" id="0" port="8000"/>
   <lp ip="PERSEUS" id="1" port="8001"/>
   <singlemap lp="0" pageid="left"/>
   <singlemap lp="1" pageid="right"/>
  </toolspecific>

 <page id="left">
  <instance id="aveLight"
       ref="../../pnmls_nets/TLC/Light.xml#Light"/>
  <instance id="strLight"
       ref="../../pnmls_nets/TLC/Light.xml#Light"/>
  <instance id="ambulance"
       ref="../../pnmls_nets/TLC/Ambulance.xml#Ambulance">
    <importPlace parameter="d_appr" instance="controller"
       ref="r_appr"/>
    <importPlace parameter="d_beforeA" instance="controller"
       ref="r_beforeA"/>
    <importPlace parameter="d_beforeS" instance="controller"
       ref="r_beforeS"/>
    <importPlace parameter="d_afterA" instance="controller"
       ref="r_afterA"/>
    <importPlace parameter="d_afterS" instance="controller"
       ref="r_afterS"/>
  </instance>
 </page>

  <page id="right">
   <instance id="controller"
        ref="../../pnmls_nets/TLC/Controller.xml#Controller">
     <importPlace parameter="d_ALG" instance="aveLight"
        ref="r_toG"/>
     <importPlace parameter="d_ALY" instance="aveLight"
        ref="r_toY"/>
     <importPlace parameter="d_ALR" instance="aveLight"
        ref="r_toR"/>
     <importPlace parameter="d_SLG" instance="strLight"
        ref="r_toG"/>
     <importPlace parameter="d_SLY" instance="strLight"
        ref="r_toY"/>
     <importPlace parameter="d_SLR" instance="strLight"
        ref="r_toR"/>
   </instance>
  </page>
 </net>
</pnml>
```

Listing 2: PNML file for model deployment and partitioning

The mapping between pages and physical computing nodes is defined in the first part of Listing 2 delimited by the tag <toolspecific>. Here, a list of <lp> tags associates each node identifier with a pair (Internet address, port) of the relevant physical node and then a list of <singlemap> tags establishes the mapping between pages and nodes.

At simulation start-up a DEVS component, named TPNDEVSDeployer, is created on a computing node and it is feed with Listing 2 file and with files defining the single modules. TPNDEVSDeployer is in charge of parsing these files and of creating an in-memory representation of each page. After the parsing phase is completed, TPNDEVSDeployer creates as many instances of SubnetManager component, as there are computing nodes. Each SubnetManager receives the representation of the pages it has to handle and thereafter it is migrated on the relevant node. Finally, when a SubnetManager reaches its destination, it creates the DEVS components corresponding to the local transitions and instantiates the data structures corresponding to places. In the case the postset of a transition resides on a different node, for each place of this postset a corresponding ReferencePlace object is created. ReferencePlace objects are responsible of transparently notifying the effect of a transition firing to the SubnetManager where actual places of the postset reside.

A transducer (statistical) object is used in the first theatre to follow the firing of transitions of lights and ambulance, as well as to check marking of relevant places. The transducer has a fire() method that gets called on each transition firing, receiving the transition id and the current time. Simulation experiments were carried out using a simulation time limit of 10^6 for each run. Listing 3 shows a portion of the method fire() checking TLC properties. For brevity, only the most important properties are shown. The places object is an hash map for retrieving a place on the local theatre from its name.

```
public void fire(String id, long now) {
     Place aveR=places.get("aveLight.toR");
     Place aveY=places.get("aveLight.toY");
     Place aveG=places.get("aveLight.toG");
     Place strR=places.get("strLight.toR");
     Place strY=places.get("strLight.toY");
     Place strG=places.get("strLight.toG");
     if( (strY.getMarking()>0 || strG.getMarking()>0) &&
         aveR.getMarking()==0 )
             log.put("Traffic allowed on both directions!! @"+now);
     if( (aveY.getMarking() > 0 || aveG.getMarking() > 0) &&
         strR.getMarking() == 0 )
             log.put("Traffic allowed on both directions!! @"+now);
     if(id.equals("ambulance.BefA") || id.equals("ambulance.BefS"))
       {
             if(strR.getMarking() == 0 || aveR.getMarking() == 0)
                    log.put("Intersection not safe at before @"+now);
             beforeT=now;
             if( id.equals("ambulance.BefA") ) beforeA = true;
             else beforeS = true;
     }
     if( (id.equals("aveLight.R2G") ) && beforeA ) {
             beforeA = false;
             long v=now-beforeT;
             if( v>max ) max=v;
             if( max>3 ) log.put("Avenue light should be green!!
  Deadline missed @"+now);
     }
     if( (id.equals("strLight.R2G") ) && beforeS ) {
             beforeS=false;
             long v=now-beforeT;
             if(v>max) max=v;
             if( max>3 ) log.put("Street light should be green!!
  Deadline missed @"+now);
     }
     ...
}//fire
```

Listing 3: A fragment of method fire() for property checking

All the previously stated properties of the TLC model were found satisfied. Therefore, from the viewpoint of simulation, the system was found to be temporally correct.

8. CONCLUSIONS

This paper extends previous work carried out by the authors and concerning an achievement with ActorDEVS of the runtime support of Time Petri Nets (Cicirelli *et al*., 2007b), i.e. an application where it is required to dynamically handle conflicting components. The proposed approach aims at providing distributed simulation for property analysis of large TPN models of time-dependent systems.

The approach proceeds according to the following steps:

- first a TPN model is graphically designed in the context of the TPN Designer toolbox (Carullo *et al*., 2003; Cicirelli *et al*., 2007c)
- then a PNML version of the model is generated where distinct sub nets are associated with distinct PNML modules. A module interface publishes a set of import/export reference places whose connection is responsibility of the configuration process. Module boundaries can be conveniently exploited for partitioning so as to fulfil local semantics requirements of transitions
- a partitioned PNML model is then parsed, instantiated and deployed on a certain number of computing nodes of the Theatre/HLA architecture, in the presence of a conservative simulation conflict-aware control engine.

On-going work is geared at:

- improving the PNML generation process from TPN Designer. At the present time the generation occurs in two steps. First TPN Designer generates an XML version of the model (externalization). Then a stylesheet is used for converting the achieved XML into PNML. The goal is to extend TPN Designer in order to produce directly the PNML
- tuning the distributed simulation infrastructure based on HLA to Pitch pRTI 1516 (Pitch) product.

REFERENCES

Billington, J., Christensen, S., van Hee, K., Kindler, E., Kummer, O., Petrucci, L., Post, R., Stehno, C. and Weber, M., 2003. The petri net markup language: concepts, technology, and tools. *Proceedings of the 24th Int. Conf. on Application and Theory of Petri Nets*, LNCS 2679, pp. 483–505. Springer.

Carullo, L., Furfaro, A., Nigro, L. and Pupo, F., 2003. Modelling and simulation of complex systems using TPN DESIGNER. *Simulation Modelling Practice and Theory*, 11 (7-8), 503-532.

Cicirelli, F., Furfaro, A., Giordano, A. and Nigro, L., 2007a. An Agent Infrastructure for distributed simulation over HLA and a case study using unmanned aerial vehicles. *Proceedings of 40th Annual Simulation Symposium (ANSS'07)*, pp. 231-238. March 26 – 28, Norfolk (VA, USA).

Cicirelli, F., Furfaro, A. and Nigro, L., 2007b. Conflict management in PDEVS: An experience in modelling and simulation of time Petri nets. *Proceedings of Summer Computer Simulation Conference (SCSC'07)*, pp. 349-356. July 15-18, S. Diego (CA, USA).

Cicirelli, F., Furfaro, A. and Nigro, L., 2007c. Using TPN/Designer and Uppaal for modular modelling and analysis of time-critical systems. *Int. J. of Simulation Systems, Science & Technology*, 8 (4), Special Issue on: Frameworks and Applications in Science and Engineering, 8-20.

Cicirelli, F., Furfaro, A. and Nigro, L., 2008. Actor-based Simulation of PDEVS Systems over HLA. *Proceedings of 41st Annual Simulation Symposium (ANSS'08)*, pp. 229-236. April 14–16, Ottawa (Canada).

DMSO, 2008. HLA-RTI, *Defense Modeling and Simulation Office*, http://www.dmso.mil/public/transition/hla. [accessed on April 2008]

Furfaro, A. and Nigro, L., 2007. Modelling and schedulability analysis of real-time sequence patterns using Time Petri Nets and Uppaal. *Proceedings of Int. Workshop on Real Time Software (RTS'07),* pp. 821-835. October 16, Wisla (Poland).

Kuhl, F., Dahmann, J. and Weatherly, R., 2000. *Creating computer simulation systems: An introduction to the High Level Architecture.* Prentice Hall.

Merlin, P. and Farber, D., 1976. Recoverability of communication protocols – implications of a theoretical study. *IEEE Transactions on Communications*, 24 (9), 1036–1043.

Murata, T., 1989. Petri nets: properties, analysis and applications. *Proc. of the IEEE*, 77 (4), 541-580.

Pitch pRTI 1516, Pitch Kunskapsutveckling AB, http://www.pitch.se/prti1516/default.asp.

Raju, S.C.V., and Shaw, A.C., 1994. A prototyping environment for specifying and checking Communicating Real-time State Machines. *Software–Practice and Experience*, 24 (2),175–195.

Zeigler, B.P., Praehofer, H. and Kim, T.G., 2000. *Theory of modeling and simulation.* 2nd edition, New York: Academic Press.

Zeigler, B.P. and Sarjoughian, H.S., 2003. *Introduction to DEVS modelling and simulation with Java: developing component-based simulation models. http://www.acims.arizona.edu*. [accessed on April 2008]

HYBRID APPROACH FOR MODELING AND CONTROL WAREHOUSE SYSTEMS

Filippo Sarri[(a)], Rinaldo Rinaldi[(b)]

[(a)(b)] University of Florence, Dipartimento di Energetica "Sergio Stecco", Via Cesare Lombroso 6/17, 50134 Firenze

[(a)] filippo.sarri@siti.de.unifi.it, [(b)] rinaldo.rinaldi@unifi.it

ABSTRACT

The aim of this paper is twofold: firstly we define a dynamical model for representing warehouse's product flows, taking into account both customer demands and stocked quantities. Secondly we introduce an automatic controller regulating products handling into the inventory structure, which aims to guarantee the satisfaction of customer demands as well as maintaining safety stocks. The proposed work is based on hybrid approach for dynamical system definition and control, developed using dedicated informatics tools. This tool is developed to improve optimal control for real systems, such as car traction control and chemical processes. In this paper we use this tool to approach a warehouse management problem improving an optimal control algorithm.

Keywords: warehouse management, hybrid systems, model predictive control (MPC).

1. INTRODUCTION

The interest about operation management is continuously rising, this trend gives impulse to new approaches to manage different enterprise aspects aiming to optimize the performance of the whole production cycle and increase competitiveness. In this contest techniques developed to optimize the production using planning and control techniques (PPC) may be applied (Vollmann and Barry 1997).

Recent studies focus on a new approach for PPC problem based on application of automatic control theory, defining a new problem formulation called Automatic Production Control (APC). In Wiendahl and Breithaupt (1998) the production process is modelled as a dynamical system which describes the material flows through the plant, given an analytic representation of internal dynamics in several work stations. The main issue of this approach is to define a controller connected to the model, which allows the automatic control of the production system. This technique aims to having an output flow of finished goods which satisfies customer demand, controlling at the same time works in progress into the plant.

Recently supply chain concept has been introduced in operations management: in this context the attention is extended on all transformation phases of material rows in finished goods, integrating logistic, production and information management. This approach consolidates the concept of material flow through the different phases of transformation process, including support logistic (internal and external), production phases and sale organization. According to APC problem formulation in literature several dynamical models to control supply chain are presented. These models are joined by chain structure to evidence the material flow through different resources which compose the modelled supply chain. This representation may be found also in models of production processes, where each block represents a particular work phase in terms of work in progress and product queues.

In Boccadoro and Martinelli (2006) a model of supply chain composed by a series of working sites is proposed, connecting the sites by downstream flow of materials and upstream flow of information. The control law is developed for satisfying the demand of final client synchronizing exchange of information and material within the chain. Using a linear discrete dynamical system to represent supply chain it's possible to implement optimal control laws using LQ or H-infinity approach.

In Dumbar and Desa (2005) a constrained model for supply chain introducing variable boundaries in a discrete linear model is developed, Mixed integer problem is solved to control the proposed system. This approach is more complex than the analytic formalization proposed in Boccadoro and Martinelli (2006), but allows to adopt a more complex and robust control law, named Model Predictive Control (MPC). This control theory appear to be a very powerful instrument also used to control hybrid systems, as in Balduzzi and Menga (2000) where an hybrid formulation for manufacturing production control is developed.

This paper focuses on controlling an inventory system which consists of two different stock areas crossed by flow materials. The model follows hybrid linear system specification which allows to determine a optimum control strategy for material flows. This strategy aims to minimize backlog orders satisfying the customer demand, also trying to maintain an acceptable safety stocks level. Hybrid formulation allows us to introduce logical values and constrains in the system dynamic making it more realistic. Using a dedicated toolbox to create the hybrid models and control them it's possible to modify and analyze the performance of close loop system; in the same way by setting the parameters of MPC law it's possible to test different warehouse configuration and working logics.

2. PROBLEM FORMULATION

For simplicity of comprehension we start modelling a single product class warehouse (i.e. inventory is made of an unique product); extension to more than one product classes is then proposed.

It may be assumed that the layout of the warehouse system is chain-like (Figure 1), where two different zones are highlighted, one called *SE* (service zone) where the input material is stocked for the replenishment in other zone, called high density zone (*HD)* which represents the warehouse structure, where products are picked for satisfying the customer demand.

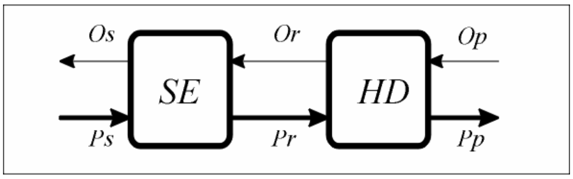


Figure 1: Warehouse layout with information flow (solid lines) and material flow (thick lines).

The *SE* and *HD* zones interact exchanging information and material, in particular the information is represented by orders: external as the customer demands or internal as the replenishment orders.

- *Op* : Product Order (customer demand)
- *Or* : Replenishment Order
- *Os* : Suppliers Order

The material flow is modelled using three variables: supplying products which enter in the system, replaced products in *HD* zone and output products picked out from the warehouse:

- *Ps* : Products from suppliers
- *Pr* : Replaced products
- *Pp* : Picked products

Warehouse dynamics are described using state equations of discrete time linear systems, where the state evolves in discrete steps, clocked by a sampling time *Ts*. The value of *Ts* is also important, because it must be compatible with data variability. Too small *Ts* raises the complexity of calculation without increasing extracted information from data, too big *Ts* reduces the sensibility of the system losing important dynamical behaviours. By using discrete notation it is possible to take advantage of the high potential of this approach, with a wide literature on modelling and controlling discrete systems, also with powerful informatics instruments for analysis and implementation. In this work we decided to represent a generic time step with *k* and the next with *k+1*, thus omitting *Ts*.

To define a correct model some hypothesis are needed regarding relations between order information and product quantities into the system. We assume that replenishment orders at generic time *k* are satisfied at *k+1*, as well as the supplier orders are satisfied with a known delay τ. Under these hypothesis we can describe the dynamics of product stocks in the high density and service zone using the following equations:

$$\begin{cases} HD(k+1) = HD(k) + Pr(k) - Pp(k) \\ SE(k+1) = SE(k) + Ps(k-t) - Pr(k) \end{cases} \tag{1}$$

New stocks in high density zone *HD(k+1)* will be equal to the quantity *HD(k)* plus replenishment products *Pr* minus the product picked out from warehouse *Pp* at the generic *k* interval. In the same way service products *SE* are increased by the products from suppliers *Ps*, and decreased by the replenishment *Pr*.

We can introduce a picking strategy related to customer demand, considering another dynamic taking into account backlog orders:

$$BL(k+1) = BL(k) + Op(k) - Pp(k) \tag{2}$$

The backlog amount *BL* depends on its previous value, customer demand *Op* and quantities delivered from the warehouse *Pp*.

The aim of this work is to implement a control algorithm that allows us to manage warehouse dynamics (1)(2) trying of obtain a material flow through the warehouse which optimizes its performance.

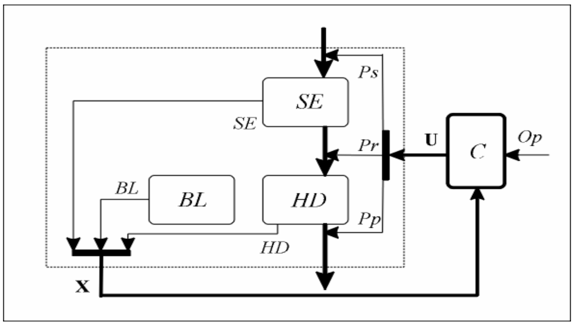


Figure 2: Control problem structure, with warehouse model (dashed block) and controller (solid block).

The control action may be applied choosing appropriate values for endogenous input of the system $\mathbf{u} = [Pr\ Ps\ Pp]^T$, depending on system state $\mathbf{x} = [HD\ SE\ BL]^T$ and exogenous input *Op* (Figure 2).

Considering a generic *k* step, given the states $\mathbf{x}(k)$ and the demand *Op(k)* we calculate the inputs $\mathbf{u}(k)$ which forces the future states $\mathbf{x}(k+1)$ to tend at the reference values $\mathbf{r}$. The control law is determined by minimization of performance index:

$$\mathbf{J}(\mathbf{x},\mathbf{u}) = \|\mathbf{Q}(\mathbf{x}-\mathbf{r})\| + \|\mathbf{Pu}\| \tag{3}$$

3. HYBRID MODEL

3.1. Dynamics

The model is generalized for a generic product class *i*, considering the dynamics and suppliers behaviour independent for each class, while constrains regard all classes depending on the warehouse structure and throughput capability.

Following the approach illustrated in the previous section we define a system state $\mathbf{x}_i = [BL_i \; HD_i \; SE_i]^T$, where:

- BL_i : Backlog orders
- HD_i : High density area stocks level
- SE_i : Service level area stocks level

an input array $\mathbf{u}_i = [o_i \, p_i \, r_i \, s_i]^T$ with:

- o_i : Customers demand
- p_i : Picked products
- r_i : Replaced products
- s_i : Product from suppliers

and system output $\mathbf{y}_i = [p_i]$. These variables are used to write the difference equations which characterize the system behaviour:

$$S : \begin{cases} BL_i(k+1) = BL_i(k) + o_i(k) - p_i(k) \\ HD_i\,(k+1) = HD_i(k) + r_i(k) - p_i(k) \\ SE_i(k+1) = SE_i(k) + s_i(k-\tau_i) - r_i(k) \\ y(k) = p_i(k) \end{cases} \tag{4}$$

The following figure illustrates a graphical form for the system described by (4):

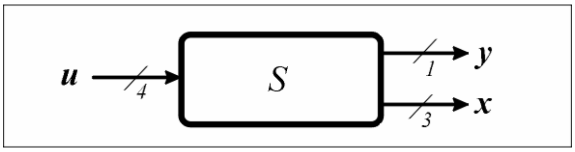


Figure 3 : Graphical representation of warehousing system *S* described by (4).

3.2. Constrains

The present approach for hybrid model structure follows the framework defined in Bemporad and Morari (1999). This work provides an useful approach for hybrid systems formulation, providing mathematical tools for studying them. This formulation allows us to add constrains for system variables, obtaining a theoretic model which simulates the behaviour of a real system.

The introduction of constraints about capacity or maximum throughput of warehouse is implemented with inequality on all *P* product class variables. We identify 3 types of constrains: structural, logistic and operative.

Structural constrains consist in the total quantities of products stored into different warehouse zones, at each step *k* these values can't exceed the maximum capability of each area, HD_{max} e SE_{max} :

$$\begin{aligned} &\sum_{i=1}^{P} HD_i(k) \le HD_{max} \\ &\sum_{i=1}^{P} SE_i(k) \le SE_{max} \end{aligned} \tag{5}$$

We consider that each product class has its own supplier, with given lead time and lot size specifications. For this reason logistic constrains to define a fixed lot size supplies and the delay of τ_i steps are introduced. Lot sizes Ls_i are inserted into the model as parameters:

$$s_i(k) \in [0, Ls_i] \qquad \forall i \in [1{:}P] \tag{6}$$

The input s_i can assume only one of two values at each time step *k*, according the fact that supplier of *i* product is activated or not. For modeling this supplying phase it's necessary to introduce a logical variable used to activate the supplier shipment, also an auxiliary real variable is needed to represent the effective product quantities introduced into the system. The introduction of logical inputs inside a continuous dynamics makes the model hybrid thus imposing the use of dedicated tools.

Operative constrains simulate the effective throughput capability of warehouse in terms of product units moving during a step, called the maximum throughput TP_{max} :

$$\sum_{i=1}^{P} r_i(k) + \sum_{i=1}^{P} p_i(k) \le TP_{max} \tag{7}$$

3.3. Mixed Logical Dynamical models

How showed in Bemporad an Morari (1999) it' s possible to define hybrid dynamics with physical laws, logical rules and constrains. An important result is the demonstration that difference equations, logics rules and inequalities can be written in an uniform form by using Mixed Integer Linear Inequalities containing continuous and logical variables. For clarity we introduced the distinction between a continuous part containing variables and equations which describe temporal dynamic, and logical part composed by boolean variables and logical rules. The connections between the two parts and an uniform representation are implemented with the introduction of auxiliary variables (continuous and logical) which allow us to write a good representation for the whole system; but on the other hand it increases the model dimensions, making control law calculation more complex.

Proposed model for warehouse system composed from dynamic (4) and constrains (5)(6)(7) could be represented by a Mixed Logical Dynamical model (MLD model) using linear equations and inequalities :

$$\begin{cases} \mathbf{x}(k+1) = A\mathbf{x}(k) + B_1\mathbf{u}(k) + B_2\boldsymbol{\sigma}(k) + B_3\mathbf{z}(k) \\ \mathbf{y}(k) = C\mathbf{x}(k) + D_1\mathbf{u}(k) + D_2\boldsymbol{\sigma}(k) + D_3\mathbf{z}(k) \\ E_2\boldsymbol{\sigma}(k) + E_3\mathbf{z}(k) \leq E_1\mathbf{u}(k) + E_4\mathbf{x}(k) + E_5 \end{cases} \tag{8}$$

the state vector, input and output contain logical and continuous variables, $\mathbf{x}(k) = [\mathbf{x}_c\ \mathbf{x}_l]$, $\mathbf{u}(k) = [\mathbf{u}_c\ \mathbf{u}_l]$, $\mathbf{y}(k) = [\mathbf{y}_c\ \mathbf{y}_l]$. The vector **z** contains auxiliary continuous variables, while **σ** contains logical ones.

The proposed model is characterized by a state with 9 continuous components :

$$\mathbf{x} = [BL_1\ HD_1\ SE_1\ BL_2\ HD_2\ SE_2\ BL_3\ HD_3\ SE_3] \tag{9}$$

representing backlog orders, products stocked in high density zone and in service zone for each class. The input is composed by 6 continuous components and 3 logical ones:

$$\mathbf{u} = [p_1\ r_1\ p_2\ r_2\ p_3\ r_3\ \mathrm{s}_1\ s_2\ \mathrm{s}_3] \tag{10}$$

representing picked and replaced quantities (*p* and *r*) and the boolean variable which activates the supplier orders (*s*) for each product class. Also we need to introduce 3 auxiliary continuous variables to convert the logical value of *s* to fixed lot quantities.

The introduction of constrains and complex dynamics exponentially increases the problem complexity, raising the number of dependent variables and the dimension of the MLD model (8). A correct analysis to reduce dynamics and constrains used for modelling every system is needed, to limit the model size and make the proposed method feasible.

4. OPTIMAL CONTROL

4.1. Feedback Structure

The aim of this method is the implementation of an algorithm which controls material flows for the model defined in the previous section. The problem is approached as a state regulation for dynamical systems, where customer demands are assimilated to state perturbations, and the controller operates forcing the state around a reference.

Focusing on a generic product class representation (Figure 3) it' s possible to structure the control problem following the next scheme:

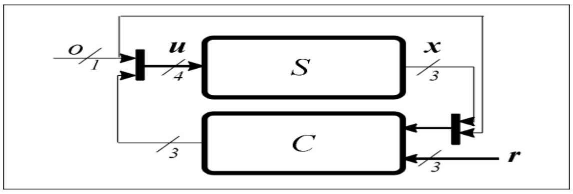


Figure 4: Control scheme with feedback connection between system *S* and controller *C*.

Figure 4 shows the feedback connection between system S and controller C. For a generic step *k* the controller C has the state system **x***(k)* and customer demand *o(k)* as inputs. C calculates the **u***(k)* signals for controlling future state **x***(k+1)* toward the constant reference **r**.

4.2. Model Predictive Control

To define control signals we used the Model Predictive Control law (MPC) that proved to be a good tool for working with hybrid systems (Bemporad and Morari 2001). This technique needs a dynamical model for the predictive approach, using a MLD model it’s possible to predict the future behaviour of the system and calculate control signals. The predictions are used to estimate the dynamics of the system during a finite period in the future (prediction horizon of *T* steps), starting from current state. The resolution of optimal control problem with MPC approach give, at generic step *k*, a sequence of *T* input signals **U***(k) which drives the state around a reference:

$$\mathbf{U}^*(k) = [\mathbf{u}^*(k)\ \ \mathbf{u}^*(k+1)\ \ldots\ \mathbf{u}^*(k+T-1)] \tag{11}$$

Only the first sample **u****(k)* of this sequence is applied to the model, discarding the others. At the next step the algorithm is replicated starting from new state observations. The receding horizon approach allow us to share the action control over several steps, satisfying the model constrains.

Optimal input sequence (11) is calculated solving a Mixed Integer Linear Programming problem (MILP), where a functional cost is minimized under constrain of a MLD dynamic. In this case to determine MPC law Mixed Integer Predictive Control problem is solved, where a performance index is minimized to obtain an optimal control sequence:

$$\mathbf{U}^*(k) = \underset{\mathbf{U}(k)}{arg\,min}\,\mathbf{J}(\mathbf{x}(k), \mathbf{U}(k)) \tag{12}$$

In the cost function **J** predicted quantities are used, where **x***(k+i|k)* represents the state estimation at step *k+i* calculated from observations at time *k* using MLD model:

$$\mathbf{J}(\mathbf{x}(k), \mathbf{U}(k)) = \sum_{i=0}^{T-1} \left\| \mathbf{x}(k+i\,|\,k) - \mathbf{r} \right\|_{\mathbf{Q}} + \left\| \mathbf{u}(k+i) \right\|_{\mathbf{P}} \tag{13}$$

This methods allow us to define an optimal control sequence for the steps between *k* and *k+T-1*.

4.3. Performance Index

Aim of this work is to adapt the illustrated control approach to modelling inventory management strategies, where the total cost function (13) can be seen as an overall performance indicator for the system. The control law is implemented to define product quantities exchanged between different areas of warehouse (10).

Considering the model (8) containing the dynamics defined in (4), it's possible to create a performance index which uses the product quantities in *SE* and *HD* zones, with particular attention to backlog orders as measure of inventory management efficiency. The quantities of products into the warehouse zones are compared to given references in order to satisfy safety stocks in the warehouse, whereas backlog orders are forced to zero to satisfy customer demands. The differences between state and references are introduced into the performance index:

$$\mathbf{J}(\mathbf{x}(k),\mathbf{U}(k))=\sum_{i=0}^{T-1}\left\|\mathbf{x}(k+i\,|\,k)-\mathbf{r}\right\|_{\mathbf{Q}} \qquad (14)$$

where **x** is the predicted state of the system, in term of backlog orders and products stocked in the zones (9).

The reference vector contains the constant expected values for each state components with zero for backlog orders and safety stock level for service and high density zones:

$$\mathbf{r} = [0\ RH_1\ RS_1\ 0\ RH_2\ RS_2\ 0\ RH_3\ RS_3] \qquad (15)$$

Q is a diagonal matrix which contains the weight of every cost component, choosing appropriate values for each component it's possible to characterize the behavior of the system, in this case we penalize backlog orders more than product level into zones. The performance index (14) is computed using infinity norm, which extracts the maximum values from a weighted vectors, as example a generic vector **v** are considered:

$$\left\|\mathbf{v}\right\|_{\mathbf{Q}} = \max_i \left|(\mathbf{Qv})_{\mathbf{i}}\right| \qquad (16)$$

where $(\mathbf{Qv})_i$ is the *i* components of row-column product between the **Q** matrix and vector **v**.

5. IMPLEMENTATION

5.1. Hysdel Language

Several representations for hybrid systems are proposed in literature (Heemels and Schutter 2001), this large number depends on a different application field which requires a dedicated form for hybrid systems. In this section is shortly illustrated the structure of Discrete Hybrid Automata (DHA), where a series of continuous dynamics influence the state switches into a finite state automata. The DHA is composed by four subparts:

Finite State Machine (FSM): that contains the discrete part of system, composed by discrete state and logical rules which sets the current state.

Switch Affine System (SAS): part containing the continuous part of the system, it's structured as a series of continuous dynamics which are selected by a dedicated index.

Event Generator (EG): this subpart allows to translate continuous information into discrete ones with generation of events, these events are generated from the analysis of continuous state in SAS.

Mode Selector (MS): it's the counterpart of EG, where discrete signals from FSM are elaborated for selecting the active dynamic in SAS.

The tool used in this paper for hybrid system definition is the HYSDEL (HYbrid Systems DEfinition Language) (Torrisi and Bemporad 2004), this application allows us to write a text representation for dynamical behaviour of systems in a well posed syntax using DHA representation.

First part of HYSDEL model is the interface, where the model structure is defined, with numeric parameters, states, inputs and outputs. Developed warehouse model works with 3 product classes each of them with dedicated suppliers. Each supplier has a known lead time, we consider class 1 is supplied with lead time of one step, class 2 with lead time 2 and third class has 3 steps of lead time. Section relative to third product class is reported below:

```
INTERFACE  {
    PARAMETER {
               REAL HD_min;
               REAL HD_max;
               REAL SE_min;
               REAL SE_max;
               REAL BL_min;
               REAL TP_max;

               REAL Ls_3;
               }
    STATE     {
               REAL BL_3 [-1e3,1e3];
               REAL HD_3 [-1e3,1e3];
               REAL SE_3 [-1e3,1e3];
               REAL se3_d1 [-1e3,1e3];
               REAL se3_d2 [-1e3,1e3];
              }
    INPUT     {
               REAL o_3 [-1e3,1e3];
               REAL p_3 [-1e3,1e3];
               REAL r_3 [-1e3,1e3];
               BOOL s_3;
              }
    OUTPUT    {
               REAL y_3;
               }
```

We can note that the input and state vectors are the same of system (4), in the state are present of two auxiliary states (*se3_d1*, *se3_d2*) needed to introduce delay for modelling supplier lead time. In interface section parameters necessary for constrain definitions related to (5)(6)(7) are defined.

Second section of HYSDEL file contains the dynamic implementation, with the definition of auxiliary variables to represent complex behaviours, and difference equations for continuous dynamics of system:

```
IMPLEMENTATION {
    AUX    {
           REAL z_s3;
           }
    DA      {
       z_s3 = {IF s_3 THEN Ls_3 ELSE 0};
           }
    CONTINUOUS
           {
           BL_3 = BL_3 + o_3 - p_3;
           HD_3 = HD_3 + r_3 - p_3;
           SE_3 = SE_3 + se3_d1 - r_3;
           se3_d1 = se3_d2;
           se3_d2= z_s3;
           }
    OUTPUT
           {
           y_3 = p_3;
           }
```

The DA section containing the expression to convert the logical input *s_3* to a product quantity, using the system parameter *Ls_3* and the auxiliary real variable *z_s3*.

The last section is used to define constrains on system variables:

```
MUST   {
       HD_1+HD_2+HD_3 <= HD_max;
       HD_1+HD_2+HD_3 >= HD_min;

       SE_1+SE_2 +SE_3<= SE_max;
       SE_1+SE_2 +SE_3>= SE_min;

       BL_3 >= BL_min;

       p_1+p_2+p_3+r_1+r_2+r_3<= TP_max;

       p_3>=0;
       r_3>=0;

       HD_3>=0;
       SE_3>=0;
       BL_3>=0;
        }
```

Used tools allows us to convert HYSDEL model in some different representations for hybrid systems, this is possible thanks to the result in Heemels and Schutter (2001), where the equivalence between different representations is showed. In this case the HYSDEL model is converted in a MLD model. The report of this operation is showed:

```
MLD  hybrid  model  generated  from  the
HYSDEL file <wh_system.hys>

12 states  (12 continuous, 0 binary)
12 inputs  (9 continuous, 3 binary)
3 outputs  (3 continuous, 0 binary)

3 continuous auxiliary variables
1 binary auxiliary variables
31 mixed-integer linear inequalities
```

5.2. MPC toolbox

Control of warehouse model is implemented with a Matlab toolbox presented in Bemporad (2004), which allows to apply a MPC law on hybrid systems. This instrument contains a command that builds a MILP problem starting from a MLD model and the parameters necessary for defining a performance index. The parameters of this function are the state components that must appear in performance index, the references for this components and coefficients for weighting each difference between state value and references. In the proposed approach only the system state contributes to the cost function (14) whose components which describe system dynamics (9), these components are weighted with a 9x9 diagonal matrix **Q**.

The MILP solution gives us a controller generating a number of signals equal to the number system input, in this model we present 3 exogenous inputs (product demands) which influence the dynamic but can't be manipulated. To avoid this problem the control law is calculated on an extended model, where exogenous inputs are assimilated by state components: this solution allows us to obtain a control input with 9 components (10) for the warehouse model.

The prediction horizon is set to 3 steps, this values allow us to consider dynamical behaviours of the system, at the same time to reduce the calculation complexity with the reduction of MILP problem dimensions. Report of MILP formulation is showed below:

```
Hybrid  controller  based  on  MLD  model
wh_extended <wh_extended.hys> [Inf-norm]

 15 state measurement(s)
 0 output reference(s)
 0 input reference(s)
 9 state reference(s)
 0  reference(s)  on  auxiliary  continuous
z-variables

 42 optimization variable(s)
(30 continuous, 12 binary)
201 mixed-integer linear inequalities
sampling time = 1, MILP solver = 'glpk'
```

The entire state contains the real state of the system (9 components), 3 auxiliary states to model supplier lead time (one for class 2 and two for class 3) and 3 extended states for product demands. The auxiliary and extended components of the state are not forced with references, because don't give additional information about inventory behaviour.

After the controller definition it's possible to simulate the behaviour of controlled model using Simulink. This operative step needs the definition of simulation run length and the product demand for each class.

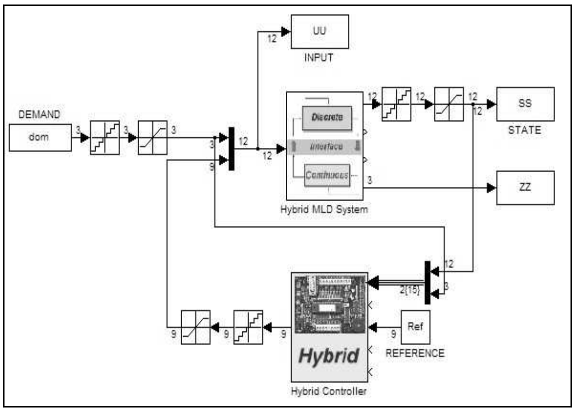


Figure 5: Simulink model to implement hybrid MPC on warehouse MLD model.

In the Simulink model (Figure 5) quantization blocks are introduced to guarantee integer quantities for state and input signals in the system. This expedient allow us to introduce more reality but increase uncertainty in the dynamics, changing the calculated quantities by the system and control blocks: this perturbation may be assimilated to a measurement noise. For this reason the infinity norm is used, to decrease the noise sensibility of closed loop system and at the same time increase the robustness of control actions.

6. NUMERICAL EXPERIMENTS

The control implementation and simulations of proposed model are developed using MATLAB 7.1 and Simulink 6.3 necessary to use the Hybrid Toolbox V 1.1.7 which requires the Model Predictive Control Toolbox.

Initially we tested the system using an impulse input for customer demands; this test highlights the behaviour of the system, showing the functioning in term of product quantities. The second test is developed to test the sensibility of the model changing the maximum throughput in HE zone.

The constant values of some parameters used during the tests are showed in Table 1:

Table 1 : Constant parameters used during the tests

	Class 1	Class 2	Class 3	Total
Ref. HD	100	150	200	450
Ref. SE	100	150	200	450
Lot Size	100	200	300	-
W_{BL}	20	20	20	-
W_{HD}	3	3	3	-
W_{SE}	1	1	1	-

For each class the reference quantities are reported, these values form the vector **r** in cost function (15), also it used as initial conditions for the simulations. The weight matrix **Q** is diagonal with non zero elements $[\ W_{BL1}\ W_{HD1}\ W_{SE1}\ W_{BL2}\ W_{HD2}\ W_{SE2}\ W_{BL3}\ W_{HD3}\ W_{SE3}]$. The chosen values give the priority at demand satisfaction during the control action, successively safety stocks are restored.

6.1. Impulse Response

For this simulation the maximum throughput of HD zone is set to 300 units for period, and customer demand is a impulse at time 1 of 400 units for each class. This test put under stress the closed loop system that works for several steps to restore the initial conditions. For simplicity the sum of all class dynamics are proposed:

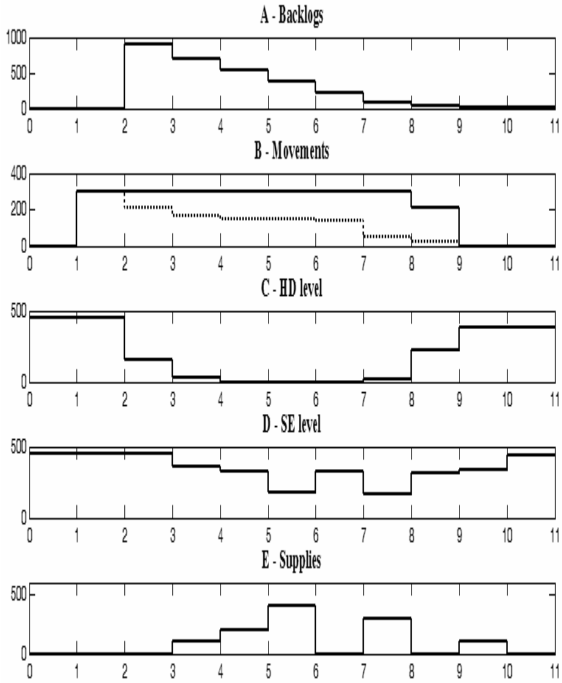


Figure 6: A - total backlog orders; B - total movement actions in HD zone (picking activity in dashed line); C - product level in HD zone; D - product level in SE zone; E - supply input of products.

The system reacts to the impulse with a step of delay for the strictly causal nature of dynamics, in Figure 6.A we can see the backlog increment at time 2. The advance reaction showed in Figure 6.B is a consequence of Simulink model (Figure 5), where the input signals are extracted with customer demands (block UU on the top) canceling the feedback delay in the graph.

Initially the controller forces the system on the picking action (dashed line in Figure 6.B) to reduce backlogs (Figure 6.A), but this behavior takes to zero the HD level (Figure 6.C), when HD zone becomes empty the replenishment quantity compensates the picking action (step 4 to 7), after that the controller increment replenishment input to restore the initial conditions. At the same time supply inputs (Figure 6.E) are set to guaranties safety stock in SE zone (Figure 6.D). The product quantities entered in the SE zone Figure 6.E) are in step form because sum of two level signals.

6.2. Throughput Analysis

This experiment aims at testing the sensibility of the proposed model changing crucial system parameters, the proposed analysis focus on maximum throughput. The movement capacity in HD zone is an important value which influences the model capability to satisfy customer demand through picking and replenishment actions.

To obtain comparable data the customer demand foe each class is maintained the same during all simulation tests :

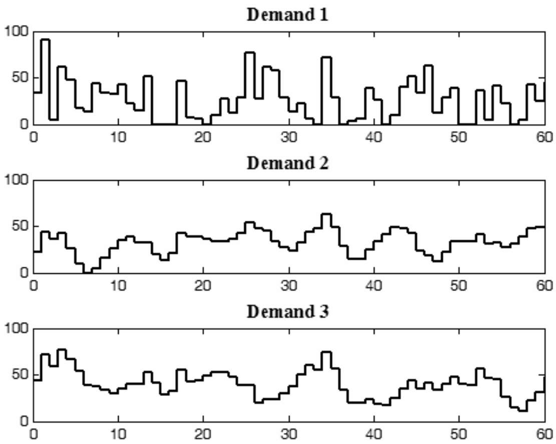


Figure 7: Demand profiles used in the throughput analysis

To evaluate the system performance we used two indexes. The first one taken into account is the percentage of backlog units over the period of simulation. This value is calculated for each class considering the ratio between the total backlog orders and total product demanded over the simulated period; with the same method the percentage of total backlogs is calculated. Second index evaluates the ratio between the handling operations in HD zone and the total handling operations capability.

These indexes are then evaluated in different simulation runs where the maximum throughput changes value between 200 and 400 units per period step. Results of these tests are showed in the next table:

Table 2: Results of throughput analysis

		Backlogs %				
	TP_m	Class 1	Class 2	Class 3	Total	Tp %
1	200	29,12	23,56	19,40	23,36	98,02
2	220	15,37	12,10	10,00	12,12	89,62
3	240	11,69	9,45	6,89	9,01	82,32
4	260	9,95	8,05	5,82	7,66	76,13
5	280	8,96	7,70	5,49	7,14	70,69
6	300	8,02	6,50	4,71	6,18	65,79
7	320	7,90	5,75	4,30	5,74	61,70
8	340	7,34	5,25	3,93	5,27	58,07
9	360	7,40	5,35	3,81	5,27	54,84
10	380	6,78	5,25	3,73	5,04	51,95
11	400	6,53	4,95	3,48	4,78	49,36

From this table we draw argue some interesting observation. With small value of throughput the warehouse performance is poor, the dynamics are conditioned by high level of backlog orders and saturation of movement capability. High values for maximum throughput assures good backlog performance but with a real low utilization of the handling capacity. The medium throughput gives good performance, balancing the backlog quantity and utilization percentage. Total backlogs and throughput percentages are plotted in Figures 8 and 9.

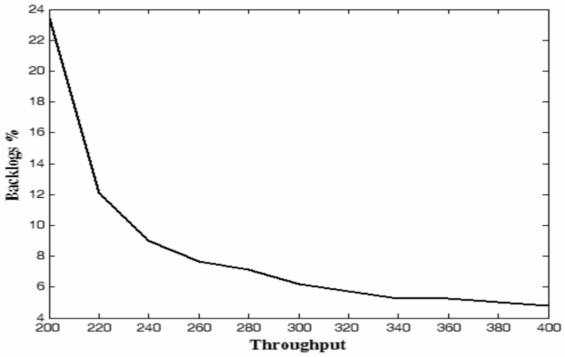


Figure 8: Percentage of total backlog units.

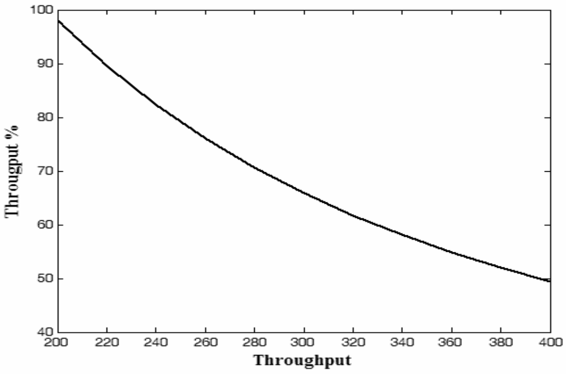


Figure 9: Percentage of capability utilization

The total results of simulation, with maximum throughput set to 300 units per period, are showed in Figure 10:

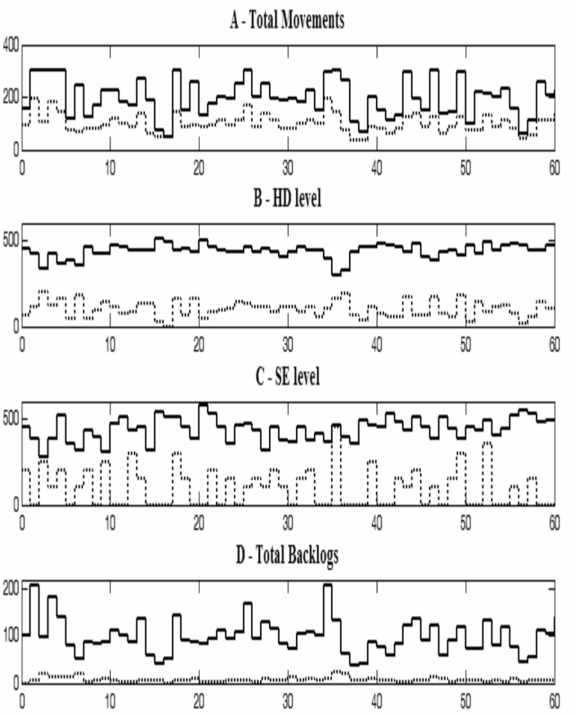


Figure 10: Total quantities of response for TP_max=300 units per period. A Total movements, picking activity in dotted line; B - HD level, replenishment activity in dotted line; C - SE level, supplies in dotted line; D - Total demand, backlog orders in dotted line.

7. CONCLUSIONS

In this paper a inventory management problem was defined using a two stages warehouse structure.

The dynamical model was developed using the hybrid system approach developed in Bemporad and Morari (1999). The HYSDEL language showed good potential to define linear hybrid system, combining easy syntax, useful code structure and powerful instruments to convert text expression in mathematical model (Torrisi and Bemporad 2004). The proposed model was characterized by different constrains for states and continuous input, moreover the use of boolean inputs allowed us to model fixed lot size supplies without significant increments of complexity. Backlog policy to manage customer demands was adopted to obtain realistic dynamics (Dumbar and Desa 2006). The decision to use a discrete time to model the warehouse, based on the availability of efficient tools, gives promising results; though a detailed analysis for parameter choice was necessary.

The MPC control policy was used to implement a inventory management strategy for the warehouse model. This algorithm was frequently adopted in control problems with constrained dynamics, in particular hybrid version (Bemporad and Morari 2001) showed high efficiency. The used toolbox (Bemporad 2004) allowed us to utilize the full flexibility of MPC law in term of cost function structure, tracking reference, weighted components and norm evaluation. The management policy adopted in this paper for inventory control gave good results about customer demand satisfactions and safety stock regulation.

The developed numerical experiments highlighted a acceptable behaviour of closed loop system, which proved to work following a logical criterion. Also the presented model demonstrates a good sensibility for parameter variations, this peculiarity increases the possibilities to use this approach in a real case study.

The proposed solution for inventory management may be extended in term of model complexity, adding logic dynamics to manage picking and replenishment phases for modelling different types of warehouse, also introducing different behaviours for product classes. The used toolbox can be used to implement very large and complex hybrid models and it give as well the instruments to analyze and control them.

REFERENCES

Balduzzi F., Menga G., 2000. Open-Loop Feedback Control for Hybrid Manufacturing Systems. *Proceedings of the 39 IEEE International Conference on Decision and Control*. December, Sydney (Australia).

Bemporad A, 2004. *Hybrid Toolbox – User's Guide.* http://www.dii.unisi.it/hybrid/toolbox.

Bemporad A., Morari M., 1999. Control of systems integrating logic, dynamics and constrains. *Automatica*, 35, pp 407-427.

Bemporad A., Morari M., 2001. Optimization-Based Hybrid Control Tools. *Proceeding of the American Conference*. June 25-27, Arlington (VA,USA).

Boccadoro M., Martinelli F., Valigi P., 2006. H-infinity control of a Supply Chain model. *Proceedings of the 45th IEEE Conference on Decision & Control*. December 13-15, San Diego (CA,USA).

Dumbar W.B.,Desa S., 2005. Distribuited Model Predictive Control for Dynamics Supply Chain Management. *International Workshop on Assessment and Future Directions of NMPC.* August 26-30, Freudenstadt-Lauterbad (Germany).

Heemels W., Schutter B. D., Bemporad A., 2001. Equivalence of hybrid dynamical model. *Automatica*, 37, pp 1085-1091.

Torrisi F. D., Bemporad A., 2004. HYSDEL – A Tool for Generating Computational Hybrid Model for Analysis and Synthesis Problems. *IEEE Transactions on Control Systems Technology*. 12(2), March.

Vollman T.E., Berry W.L., Whybark D.C., 1997. Manufacturing Planning & Control Systems. *McGraw-Hill.*

Wiendhal H.P., Breithaupt J.W., 2000. Automatic production control applying control theory. *International Journal of Production Economics*, pp 33-46.

IMPROVING A PROCESS IN A BRAZILIAN AUTOMOTIVE PLANT APPLYING PROCESS MAPPING, DESIGN OF EXPERIMENTS AND DISCRETE EVENTS SIMULATION

José Arnaldo Barra Montevechi[(a)], Alexandre Ferreira de Pinho[(b)], Fabiano Leal [(c)], Fernando Augusto Silva Marins[(d)], Rafael Florêncio da Silva Costa[(e)]

[(a), (b), (c), (e)]Universidade Federal de Itajubá - UNIFEI
[(d)]Universidade Estadual Paulista - UNESP

[(a)]montevechi@unifei.edu.br, [(b)]pinho@unifei.edu.br, [(c)]leal@unifei.edu.br, [(d)]fmarins@feg.unesp.br,
[(e)]rafael.florencio@yahoo.com.br

ABSTRACT

The objective of this article is to apply the Design of Experiments technique along with the Discrete Events Simulation technique in an automotive process. The benefits of the design of experiments in simulation include the possibility to improve the performance in the simulation process, avoiding trial and error to seek solutions. The methodology of the conjoint use of Design of Experiments and Computer Simulation is presented to assess the effects of the variables and its interactions involved in the process. In this paper, the efficacy of the use of process mapping and design of experiments on the phases of conception and analysis are confirmed.

Keywords: Design of Experiments, Discrete Events Simulation, Process Mapping

1. INTRODUCTION

As an example of conjoint application of simulation and design of experiments, the work of Nazzala, Mollaghasemi and Anderson (2006) can be cited, as it integrates simulation, design of experiments and economic analysis in a decision-making process at a semiconductors company, using a simulation model validated according to techniques presented by Law and Kelton (2000), Nayani and Mollaghasemi (1998), and Sargent (1998).

The benefits of the design of experiments in simulation include the possibility of improving the performance on the simulation process, avoiding the trial-and-error techniques to seek solutions. Kleijnen, Sanchez, Lucas and Cioppa (2005) affirm that research related to design of experiments is frequently found in specialized magazines, rarely read by simulation practitioners.

Thus, the objective of this article is to apply the design of experiments techniques along with the discrete events simulation technique in a process of an automotive industry. The effect of the input variables (number of ungrease machine, number of shifts and numbers of works) over the output variable (total of parts produced per day) is intended to be evaluated. Moreover, the effects of the interaction between the input variables through full factorial design is also intended to be assessed.

2. COMPUTER SIMULATION IN MANUFACTURING ENVIRONMENTS

According to Harrell, Ghosh and Bowden (2000), and Law and Kelton (2000), simulation is the imitation of a real system, computer-modeled, for evaluation and improvement of its performance. In other words, simulation is the importing of reality to a controlled environment where its behavior can be studied, under various conditions, without physical risks and/or large costs involved. Banks (2000) affirms that simulation involves the creation of an artificial history of reality and, based on this artificial history, observations and inferences on the operating characteristics of the real system that is represented can be made.

O'Kane, Spenceley and Taylor (2000) affirm that simulation has become one of the most popular techniques to analyze complex problems in manufacturing environments.

According to Banks, Carson, Nelson and Nicol (2005), simulation is one of the most widely used tools in manufacturing systems, more than in any other area. Some reasons may be enumerated:

- The increase in productivity and quality in the industry is a direct result from automation. As automation systems become more and more complex, they can only be analyzed using simulation;
- The costs with equipment and installations are huge;
- Computers are becoming cheaper and faster;
- Improvements on simulation software reduced the time of development of models;
- The availability of animation resulted in higher comprehension and usage from manufacturing managers.

3. DESIGN OF EXPERIMENTS

The word experiment is used in a very precise form to indicate an investigation where the system under study is under control of the investigator. On the contrary, for an observational study, some characteristics will be out of the control of the investigator (Cox and Reid, 2000). According to Montgomery (2001), the experiment can be seen as a test, or as a series of tests, in which the proposed changes are applied on the input variables of a process or system, to, then, observe and identify the changes occurred on the output variables.

Still according to Montgomery (2001), the design of experiments refers to the process of planning of experiments in a way that appropriate data can be analyzed through statistical methods, resulting in valid and objective conclusions. According to Kelton (1999), one of the main goals of the experimental design is to estimate how changes in input factors affect the results, or answers of the experiment.

Some terms are commonly used in design of experiments. Mason, Gunst and Hess (2003) define factor as a controllable experimental variable, which variation influences the response variable. Each factor must assume some values, defined as levels. The changes occurred on the mean of the values of the response variable correspond to the effects.

Besides the effects caused by the factors, the effects created by the interaction of the factors can be determined. These interactions correspond to combined effects, where the effect of each factor depends on the levels of the other factors.

The decisional process of the experimentalist falls back on a trade-off between performance versus cost, where the necessity of a more precise recognition of a greater number of factors and interactions bears a larger number of experiments and replicates.

According to Cox and Reid (2000), the advantages of the use of a full factorial lies on the higher efficiency in estimating the main effects of the factors under the variable in analysis, and, specially, the definition of the interaction among all the factors.

According to Sanchez, Moeeni and Sanchez (2006), many studies related to operations management use the full factorial design of experiments because of its simplicity and due to the fact that its project allows the analyst to identify interactions among the factors, as well as the main effects. As examples of these studies works from Enns (1995), who used the factorial design to evaluate the usage rate and sequencing techniques in a process can be mentioned. Malhotra and Ritzman (1994) considered a full factorial design to evaluate the impact of demand variability, usage capacity, and route flexibility in postal service stations.

The disadvantage of the use of the full factorial lies on the amount of time and experiments to be made. According to Kelton (1999), when the number of factors becomes moderately large, the number of experiments explodes. A possible solution for this situation is the use of fractional factorial, in which only a fraction of all possible combinations are evaluated. This solution is indicated to situations where a great number of factors to be analyzed exist, where only the main effects of the factors are considered important.

The realization of the experiments is frequently expensive or even impracticable. For these situations, the use of simulated experiments is recommended, and this integration between design of experiments and simulations is presented as follows.

4. DESIGN OF SIMULATED EXPERIMENTS

According to Kelton (1999), the use of simulation aids directly the execution of experiments that are costly or even impossible to be carried out in practice. Still illustrating the conjoint applicability of design of experiments and simulation, Barton (2002) points out in his work that, typically, simulation researchers emphasize the realization of experiments on the simulator and the analysis of the responses obtained, not paying attention to a previous phase to this one, where the planning of the experiment to be carried out must be made, with the possibility of being carried out in practice through Design of Experiments (DOE).

According to Kleijnen, Sanchez, Lucas and Cioppa (2005), many simulation practitioners could obtain more information from their analysis if they used statistical theories, more specifically with the use of design of experiments developed specifically for computer models.

To understand the role of simulation in the execution of experiments, it is only necessary to imagine that, in an experiment, the response variable (RV), or dependent variable, can be represented as: RV = f (IV), where IV represents the input variables (independent variables). Thus, according to Kelton (1999), the transformation function f represents the simulation model itself.

Nowadays, simulation software accompanies generators of random numbers. So, according to Kelton (1999), from the design of experiments point of view, it is possible to excerpt the experimental randomization that frequently corresponds to a hard problem in physical experiments. Another important verification is that, when the execution of the experiments occurs in a simulation model, all the input factors became controllable. Barton (2002) emphasizes this statement affirming that nuisance variables rarely appear in simulation models, because normally, there is control over all other factors.

Another important advantage of the use of simulation in the execution of experiments is replicating the experiments many times, obtaining many estimates of the main effects and interaction of the factors. These replicates will favor the conclusion of the significance or non-significance of the effect. Frequently, physical experiments are made in a small number of replicates, or even without them, due to the experiment difficulty or even the cost involved. These limitations do not occur in simulation once the model is constructed and validated.

Dessouky and Bayer (2002) integrated the selection of the maintenance policy in the process construction phase, through simulation (software ProModel 4.0) and application of DOE, through the Taguchi technique.

In this research, 4 factors (problem severity, maintenance policy, resource capacity and validation time) and 2 levels were defined. The experience of the specialists and the system knowledge were used to define the factors and variables.

Schappo (2006) presents an analysis of an assembly process of compressors and cellular layout alternatives, making use of design of experiments techniques and computer simulation as an analysis tool of the different scenarios with the objective of quantitatively measuring the changes introduced on the system which is being studied, improving the performance indicators in the manufacturing process as well as productivity increase.

In the opinion of Kleijnen, Sanchez, Lucas and Cioppa (2005), most projects were originally developed for experimentation in the real world, and they have been adapted to be used in simulation studies, instead of being developed specifically inside the simulation principles. Classical texts of design of experiments, such as Box, Hunter and Hunter (1978), Box and Draper (1987), Montgomery (2001), Myers and Montgomery (2002), focus not on the necessity of simulation analysts, but on the practical restrictions and conduction of experiments in the real world. Other texts about simulation such as Law and Kelton (2000) and Banks, Carson, Nelson and Nicol (2005) cover a vast array of topics on the subject; however the demonstration of design of experiments is done using simpler problems that do not stimulate the mental abstraction of readers about the deepness and application possibilities.

Therefore, it may be affirmed that practitioners of simulation must be attentive to the use of experiment projects as a necessary part of the analysis of complex simulations.

5. METHODOLOGY

According to Silva and Menezes (2005), a research is experimental when an object of study is determined, variables that would be capable of influencing it are selected, and ways of controlling and observing the effects that the variable produces on the object are defined. On this experimental research, the experiments will be carried out through a discrete events simulation.

Using an analysis of the simulation methodology proposed by Chwif (1999) and the steps proposed by Montgomery (2001) for design of experiments, a flow-chart that tries to explain the logic of the simulation process was constructed, in which the experimentation phase is conducted by DOE. This research was conducted according to the methodology shown in figure 1. In this methodology, there are three models that must be made: the conceptual model, the computer model and the operational model. In the same manner, these models must be validated or verified. According to Law (1991), the creation phase of the conceptual model is the most important aspect in a study of simulation. Chwif and Medina (2006) dedicate special attention to this model, since, according to them, many simulation researches do not demonstrate this stage. According to Shannon (1975), an effective conceptual modeling can lead to the identification of an adequate solution, avoiding the necessity of a complete simulation study. Works such as Greasley's (2006) use process mapping as a means of describing the logic and determining decision points, even before the computer model, created in the software Arena®.

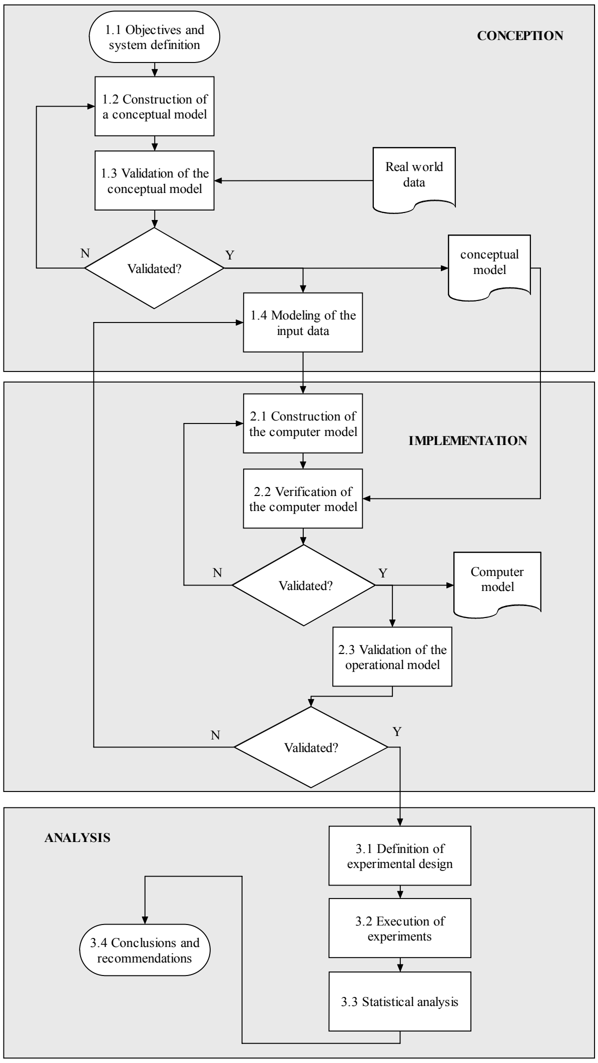

1.1 Objectives and system definition
CONCEPTION
1.2 Construction of a conceptual model
1.3 Validation of the conceptual model
Real world data
N
Validated?
Y
conceptual model
1.4 Modeling of the input data
2.1 Construction of the computer model
IMPLEMENTATION
2.2 Verification of the computer model
N
Validated?
Y
Computer model
2.3 Validation of the operational model
N
Validated?
Y
ANALYSIS
3.1 Definition of experimental design
3.2 Execution of experiments
3.4 Conclusions and recommendations
3.3 Statistical analysis

Figure 1: Methodology of Conduct of this Research
Montevechi, Pinho, Leal and Marins (2007)

In this paper, the conceptual model will be the starting point, supplying information to the computer model. The representation of the conceptual model will be made in the form of a process mapping, executed using a selected technique. According to Leal (2003) and Pinho, Leal and Almeida (2006), these techniques must be selected according to the characteristics of the process and the objectives of the work. Besides, the generated maps must not be characterized as an end, but as a feasible means for visualizing improvements.

In this research, the process map will be used, which, according to Barnes (1982), is a technique to register a process in a compact way, through some standardized symbols like operations, transports, inspections, delays and warehousing. The level of detail achieved by this technique is compatible with the objective of this research. Moreover, it is a well-known tool, which facilitates the validation process of the conceptual model, along with process specialists (real system).

The computer model is obtained through conversion of the conceptual model using some simulation language or a commercial simulator.

The simulator selected for this work is Promodel®. Hlupic and Paul (1996) present a methodological approach to select the simulation software, according to some criterion, as cost and processing time. The reasons for choosing this software for this research are due to, above all, the use of graphic animation, an important ally in the verification and validation of the model. In the same manner, works from authors such as Verma, Gibbs and Gilgan (2000) justify the use of Promodel® due to the possibility to analyze the simulation through the accompaniment of the animation, being easy to apply and interpret.

After the elaboration of the computer model, it must be verified whether its behavior is according to the conceptual model. This process is called model verification. The verification also consists of eliminating bugs from the model.

After obtaining and verifying the computer model, it must be submitted to various runs obtaining, this way, results in different scenarios. These results must be compared with the results of a real system, in order to verify the size of the error. Once the error is inside the acceptable limits, the model is apt to execute experiments, called operational model.

6. APPLICATION OF THE PROPOSED METHODOLOGY

The way the steps of the proposed methodology of the above item were conducted during this research will be described as follows.

Step 1.1: Objectives and system definition

The objective of this work is determining which variables, among the selected, have more influence over the total daily production of the system. The three analyzed variables were: number of ungrease machine, number of shifts and numbers of works. These variables were selected because they were pointed out by the specialists of the real system as strategic.

First it was chosen a case from an automotive industry. The final product of this company is a ring which is used in the automobile engines. The model that will be elaborated is classified as a discrete event simulation.

The study case is related to a manufacturing cell disposed at the beginning of the production process. This cell is responsible for the manufacture of the first part of the production process. Eight machines and two workers are used in the manufacturing cell which works along three shifts. Each worker is able to operate two machines which are used to feed other two machines and also to transport components inside and outside the cell. The workers involved in that job are also responsible for the inspection of the products and to set up each machine.

The simulation time was 18 days (three weeks), with three shifts (8 hours each).

Step 1.2: Construction of a conceptual model

In this phase, the real model under investigation is summarized using the conceptual model, which is simply a series of logical relationships relative to the components and structure of the system.

This conceptual model won't be demonstrated in this article, for solicitation of the company.

Step 1.3: Validation of the conceptual model

The validation was realized through comparison between the process mapping and the real situation. The mapping, realized by the researchers, was presented to the process specialists and to people from the company not directly related to the real system. This way, once verified that the system is correctly represented, the conceptual model is registered.

Step 1.4: Modeling of the input data

For the realization of this work, the software Promodel® from Promodel Corporation was used. It is one of the most used softwares on the market (Doloi and Jafari, 2003). This package incorporates three main programs: Promodel® (for simulation of discrete events), SimRunner® (for the optimization of optimization models), and Stat::Fit® (for probability distribution studies).

The processing times were measured in the object of study and inserted in the software Stat::Fit® with the objective of obtaining the candidate probability distributions and the adherence test.

Step 2.1: Construction of a computer model

For the construction of a model, ProModel® presents the following elements: places, entities, resources, processing and arrivals. The definitions and the functionality of the main elements are shown as follows:

Places: Represent the fixed places of the system, where the processes are carried out. These elements are used for the representation of workstations, buffers, conveyors and waiting lines. In this element, capacity, units (simple or multiple), setups, maintenance, statistical detail level and rules for the arrival and departure of materials can be defined.

Entities: The entities are items to be processed by the system, and they can be: raw material, pallets, people, or documents. The entities have defined speed, besides having statistical level like the places. They can be grouped or divided along the productive process, being moved from one place to another using a defined route or a work network.

Variables: Can be global or local. The global variables are used to represent mutable numeric values. The local variables only establish functions in the part of logic in which they are declared, and both can contain either numerical or real values. A global variable can be referred to in any place of the model. On the other hand, the local variable can only be referred to inside a determined block where it has been declared.

Attributes: Similar to the variables, the attributes are attached together with entities and specific places and usually contain information about them. They can contain real or whole numbers. From its usage only one entity referring to one type of part can be created and seven attributes to differentiate the seven types of parts to be modeled, as occurs in this paper.

Arrivals: Defines the input of the entities inside the model. The quantities, frequency, arrival periods as well as the arrival logic can be defined. Also, the arrivals through an external arrival file of parts referred to in the file editor can be defined.

Processing: Consists of a table where the operations of each entity in each place as well as the necessary resource for this operation, and a table of routes that determines the destiny and movement of each entity as well as the way in which this movement happens and the necessary resources are defined.

Resources: Are the elements used to transport entities, execute operations, and make place maintenance. They can be either people or equipment. A system can have one or more resources, endowed with movement or not. However, for each mobile resource a path network must be designated, that is, a route in which movement will happen.

Figure 2 illustrates the construction environment of the computer model from ProModel®.

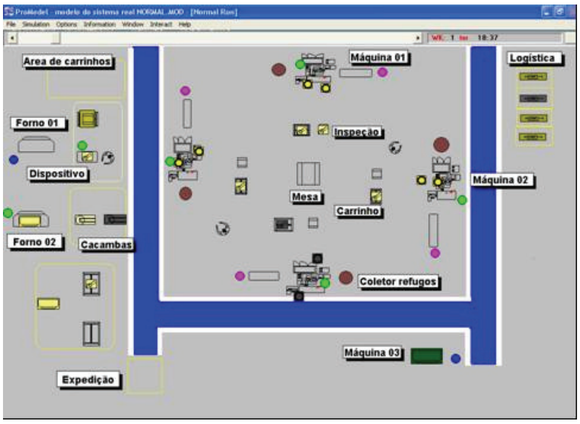


Figure 2: Environment of the Computer Model from ProModel®

Step 2.2: Verification of the computer model

This model was verified by the following procedure: the model was built in stages; twenty versions were constructed; only when the model of each stage was working correctly the new stage could be initialized.

Furthermore, at the beginning only deterministic data was used to simulate the model. This kind of data makes easier for the user to check if the logic is corrected in the model. Also, the debugger of ProModel®, the software used to perform the simulation, pointed out to the researches some mistakes. The tests runs were performed with the animation function working. This function allows the user to verify the inconsistencies in the parts flow.

After that, stochastic values were used in the computer model.

Step 2.3: Validation of the operational model.

Before using statistical techniques, a face to face verification was applied. In this verification the model was introduced to the company specialists. Using animation these specialists could evaluate the model working. In this verification the model was accepted.

In the analyzed case, the variable used for comparison represents the number of parts manufactured in one day. The statistical distribution that best fits this variable behavior is a Poisson distribution.

Bisgaard and Fuller (1994) said that in works with design of experiment the response is typically measured in a continuous scale. However, in many cases the unique economic useful measure is to count the numbers of defects or defected parts. Almost every statistical method, particularly the methods applied on the factorial experiment analysis, is based under the assumption that the response is measured in a continuous scale and has constant variance. According to the authors, derivations show that it is possible to obtain a constant variance to defects counting using square root over the collected data, in the case of a Poisson distribution.

Applying the square root function in the real and simulated data, it is possible to verify if the new data can be represented by a normal distribution.

Figures 3 and 4 show the normality test performed using the Minitab Software in the real and simulated data.

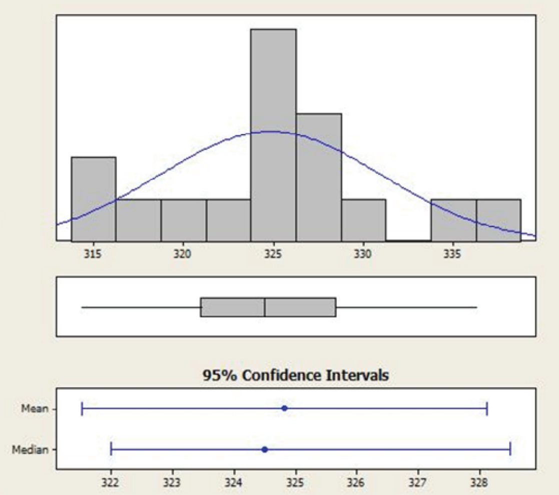


Figure 3: Anderson-Darling Normality Test for Simulation Data

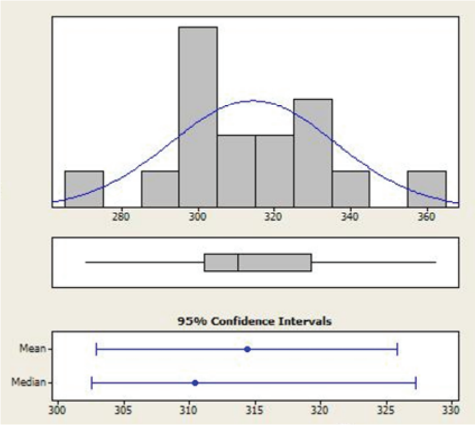


Figure 4: Anderson-Darling Normality Test for Real Data

In this test, the p-values found were 0.460 and 0.565. Both are greater than 0.05 then it is possible to consider as a normal distribution.

After the real and simulated data were considered as a normal distribution, it is possible to perform the F-test. The F-test is done comparing the variance between the system and the model data. The null hypothesis is the one which the variances of both data (from the real system and the model) are equal. The alternative hypothesis is the one which the variances of both data are not equal.

Figure 5 shows that the null hypothesis should be rejected. This test was done using Minitab.

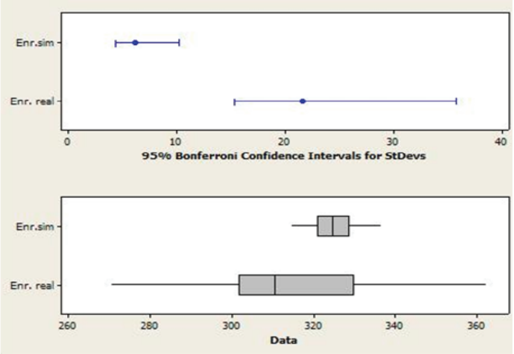
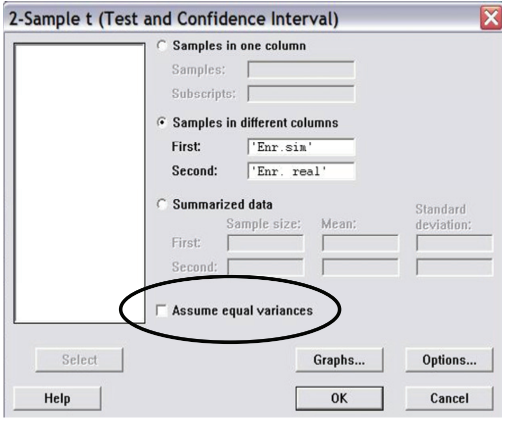


Figure 5: Variance Similar Test

Once the variances are not equal the option highlighted in the figure 6 should not be selected.


2-Sample t (Test and Confidence Interval)

○ Samples in one column

● Samples in different columns

First: 'Enr.sim'

Second: 'Enr. real'

○ Summarized data

☐ Assume equal variances

Figure 6: t-test for Dissimilar Variance

Following with the test using Minitab, the t-test is accepted since the p-value is 0.079 greater then α = 0.05. Thus the model is a good representation of the real system according to the statistical techniques.

Step 3.1: Definition of Experimental design

The factors selected for the work correspond to those the team defined as most probable of having contribution in the total amount of parts produced by the system.

Factor A: Number of ungrease machine: This factor expresses the amount of ungrease machines.

Factor B: Number of shifts: This factor expresses the amount of shifts.

Factor C: Number of workers: This factor expresses the amount of workers.

In summary, the experiment planning is as follows:

- Number of Factors: 3;
- Number of Levels: 2;
- Number of Experiments: 8;
- Number of Replicates: 3.

Table 1 shows the experimental conditions for the described experiment.

Table 1: Experimental Conditions

Factors	Level -	Level +
A:	1	2
B:	2	3
C:	2	4

Step 3.2: Execution of the Experiments

Table 2 shows the experimental matrix for the experiment planning described on step 3.1. In this table, r1, r2 and r3 mean replication 1, replication 2 and replication 3. The replications values mean total amount of parts produced by the system.

Table 2: Experimental Matrix

	A	B	C	r1	r2	r3
1	-	-	-	104000	91000	104000
2	+	-	-	104000	104000	104000
3	-	+	-	104000	104000	65000
4	+	+	-	91000	78000	78000
5	-	-	+	104000	78000	117000
6	+	-	+	91000	78000	78000
7	-	+	+	104000	104000	117000
8	+	+	+	104000	104000	104000

Step 3.3: Statistical Analysis

The analysis of the main effects of each factor, presented in figure 7, shows that factor C (number of workers) has a strong positive effect over the final response, that is, the daily amount produced. This means that the alteration of level (-) to level (+) increases the final result. That factor A (number of ungrease machine) has a strong negative effect over the final response. MiniTab® was the software used for the developed calculations.

The weight of the effects can be visualized in figure 8. On this graph, it can be verified that just BC factors interaction effect (number of shifts * number of workers) are significant for a degree of confidence of 95%, as shown by cutoff line for statistical significance.

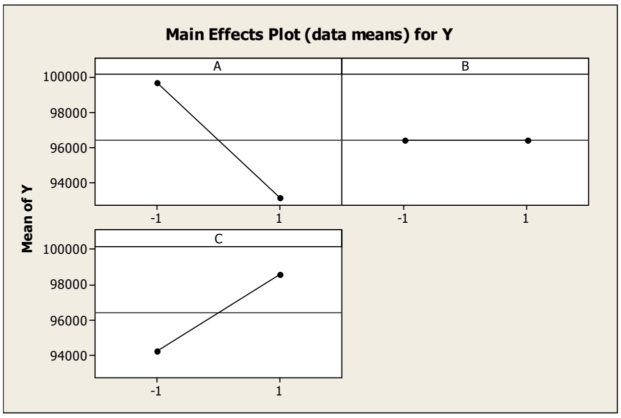


Figure 7: Graph of the Main Effects

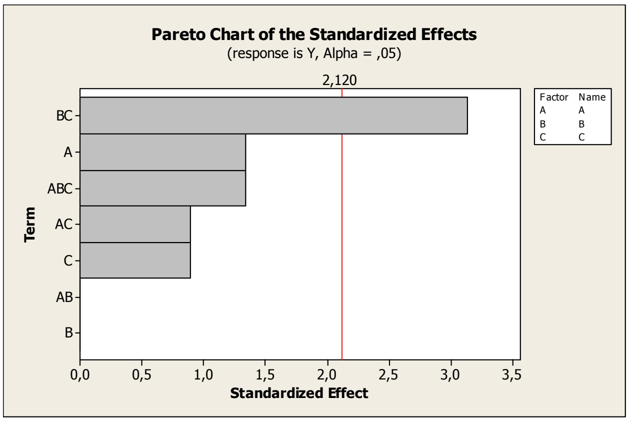


Figure 8: Pareto Chart of the Standardized Effects

Step 3.4: Conclusions and Recommendations

The factors B and C isolated do not have a significant effect on the response variable. However, as showed in the previous item, the interaction between these factors has a significant effect on the response variable. This effect could be hardly verified by specialists without the use of design of experiment.

7. CONCLUSION

The use of Design of Experiments conjointly with Computer Simulation allowed a more efficient analysis of the results of the simulated model. Moreover, the significance of the effects of the interactions were confirmed, aiding the managerial decision making process.

It is also important to emphasize that the process mapping technique made possible the creation of the conceptual model in the conception phase of the used methodology, as shown in figure 1. The same way, the Design of Experiments technique made possible a more meticulous analysis in the analysis phase.

It is good to emphasize that the experimentation phase on the simulation should be preceded by the verification and validation phases. In this current case the statistical validation guarantee that the output data from the simulated model might represent the behavior of the real system.

For the company, the recognition of the individual and combined effects of the factors favors the elaboration of an improvement plan for the increase of the daily production rate. Since the productive process is dynamic, that is, it is altered according to market demand, knowing the most relevant factors of this process facilitates the decision making, about the factor levels, by the managers.

ACKNOWLEDGMENTS

The authors acknowledge CAPES (Project 023/05) and FAPEMIG for supporting this research.

REFERENCES

Banks, J. 2000. Introduction to simulation. *Proceedings of the Winter Simulation Conference*. Atlanta.

Banks, J.; Carson, J. S.; Nelson, B. L.; Nicol, D. M. 2005. *Discrete-event Simulation*. Fourth Edition. Upper Saddle River, NJ: Prentice-Hall.

Barnes, R. M. 1982. *Estudo de movimentos e de tempos*. 6ª ed. São Paulo: Edgard Blücher.

Barton, R. R. 2002. Designing simulation experiments. *Proceedings of the Winter Simulation Conference*, San Diego, California, USA.

Bisgaard, S.; Fuller, H. 1994. Analysis of Factorial Experiments with defects or defectives as the response. Report n. 119, *Center for Quality and Productivity Improvement University of Wisconsin*.

Box, G. E. P., Draper, R. 1987. *Empirical Model-Building with Response Surfaces*. New York: Wiley.

Box, G. E. P., Hunter, W. G., Hunter, J. S. 1987. *Statistics for Experimenters: An Introduction to Design, Data Analysis and Model Building*. New York: Wiley.

Chwif, L. 1999. *Redução de modelos de simulação de eventos discretos na sua concepção: uma abordagem causal*. Tese de doutorado. Escola Politécnica da USP, Departamento de Engenharia Mecânica.

Chwif, L.; Medina, A.C. 2006. *Modelagem e Simulação de Eventos Discretos: Teoria e Aplicações*. São Paulo: Editora Dos Autores.

Cox, D.R.; Reid, N. 2000. *The Theory of the Design of Experiments*. Chapman & Hall/CRC.323p.

Dessouky, Y.M.; Bayer, A. 2002. A simulation and design of experiments modeling approach to minimize building maintenance costs. *Computers & Industrial Engineering*, 43.

Doloi, H.; Jaafari, A. 2003. Conceptual simulation model for strategic decision evaluation in project management. *Logistics Information Management*, v. 15, n. 2.

Enns, S.T. 1995. An integrated system for controlling work load and flow in a job shop. *International Journal of Production Research*, 33, 2801–2820.

Greasley, A. 2006. Using process mapping and business process simulation to support a process-based approach to change in a public sector organization. *Technovation*, n. 26, p.95-103.

Harrell, C. R.; Ghosh, B. K.; Bowden, R. 2000. *Simulation Using Promodel*. 3.ed. Boston: McGraw-Hill. 603 p.

Hlupic, V.; Paul, R.J. 1996. Methodological approach manufacturing simulation selection. *Computer Integrated Manufacturing Systems*, v. 9, n.1: p.49-55.

Kelton, W.D. 1999. Designing simulation experiments. *Proceedings of the Winter Simulation Conference,*Phoenix, AZ, USA..

Kleijnen, J. P. C.; Sanchez, S. M.; Lucas, T.W.; Cioppa, T. M. 2005. State-of-the-Art Review: A User's Guide to the Brave New World of Designing Simulation Experiments. *Journal on Computing*, vol. 17, n. 3, p. 263–289.

Law, A. 1991. Simulation model´s level of detail determines effectiveness. *Industrial Engineering*, v.23, n.10: p.16-18.

Law, A. M., Kelton, W. D. 2000. *Simulation Modeling and Analysis*. 3rd ed. New York: McGraw-Hill.

Leal, F. 2003. *Um diagnóstico do processo de atendimento a clientes em uma agência bancária através de mapeamento de processos e simulação computacional*. Dissertação (Mestrado em Engenharia de Produção) Programa de Pós-Graduação em Engenharia de Produção, Itajubá, MG, UNIFEI.

Malhotra, M. K., Ritzman, L. P. 1994. Scheduling flexibility in the service sector: a postal case study. *Production & Operations Management*, v. 3, p. 100–117.

Mason, R. L.; Gunst, R. F.; Hess, J. L. 2003. *Statistical Design and Analysis of Experiments*. John Wiley & Sons Publication.

Montevechi, J.A.B; Pinho, A, F. de; Leal, F.; Marins, F. A. S. 2007. Application of design of experiments on the simulation of a process in an automotive industry. *Proceedings of the Winter Simulation Conference*. Washington, DC, USA.

Montgomery, D. C. 2001. Design and Analysis of Experiments, 5th edition, John Wiley & Sons, Inc.

Myers, R. H.; Montgomery, D. C. 2002. *Response Surface Methodology: Process and Product Optimization using Designed Experiments*. Second edition, New York: Wiley.

Nayani, N., Mollaghasemi, M. 1998. Validation and verification of the simulation model of a photolithography process in semiconductor manufacturing. *Proceedings of the Winter Simulation Conference*, Washington, DC, USA.

Nazzala, D.; Mollaghasemi, M; Anderson, C. D. 2006. A simulation-based evaluation of the cost of cycle time reduction in Agere Systems wafer fabrication facility - a case study. *International Journal of Production Economics*, v. 100, p. 300–313.

O'Kane, J.F., Spenceley, J. R. Taylor, R. 2000. Simulation as an essential tool for advanced manufacturing technology problems. *Journal of Materials Processing Technology*, 107, 412-424.

Pinho, A. F.; Leal, F.; Almeida, D. A. 2006. A Integração entre o Mapeamento de Processo e o Mapeamento de Falhas: dois casos de aplicação no setor elétrico. *Anais do XXVI ENEGEP – Encontro Nacional de Engenharia de Produção*. Fortaleza, Ceará, Brasil.

Sanchez, S. M.; Moeeni, F.; Sanchez, P. J. 2006. So many factors, so little time: Simulation experiments in the frequency domain. *International Journal of Production Economics*. 103, 149–165.

Sargent, R. G. 1998. Verification and validation of simulation models. *Proceedings of the Winter Simulation Conference*.

Schappo, A. J. 2006. *Um método utilizando simulação discreta e projeto experimental para avaliar o fluxo na manufatura enxuta*. Dissertação submetida ao programa de Pós Graduação em Engenharia de Produção e Sistemas da Universidade Federal de Santa Catarina, Florianópolis.

Shannon, R. E. 1975. *Systems simulation: the art and science*. Englewood Cliffs: Prentice-Hall.

Silva, E. L. da; Menezes, E.M. 2005. *Metodologia da pesquisa e elaboração de dissertação*. 4.ed. Universidade Federal de Santa Catarina, Florianópolis, SC.

Verma, R.; Gibbs, G. D.; Gilgan, R. J. 2000. Redesigning check-processing operations using animated computer simulation. *Business Process Management*, v.6, n.1, p.54-64.

AUTHORS BIOGRAPHY

José Arnaldo Barra Montevechi is a Titular Professor of Instituto de Engenharia de Produção e Gestão at Federal University of Itajubá, in Brazil. He holds the degrees of Mechanical Engineer from Federal University of Itajubá and M.Sc. in Mechanical Engineer from Federal University of Santa Catarina, and Doctorate of Engineering from Polytechnic School of University of São Paulo. His research interest includes Operational Research, Simulation and Economic Engineering. His e-mail address is <montevechi@unifei.edu.br>.

Alexandre Ferreira de Pinho is a Professor of Instituto de Engenharia de Produção e Gestão at Federal University of Itajubá, in Brazil. He holds the degrees of Mechanical Engineer from Federal University of Itajubá and M.Sc. in the same university. His research interest includes Operational Research, Simulation and Operations Management. His e-mail address is <pinho@unifei.edu.br>.

Fabiano Leal is a Professor of Instituto de Engenharia de Produção e Gestão at Federal University of Itajubá, in Brazil. He holds the degrees of Mechanical Engineer from Federal University of Itajubá and M.Sc. in the same university. His research interest includes Simulation, Operations Management and Work Study. His e-mail address is <fleal@unifei.edu.br>.

Fernando Augusto Silva Marins is a Professor of Engineering College – Campus of Guaratinguetá – São Paulo State University, in Brazil. He holds the degree of Mechanical Engineer from São Paulo State University, the M.Sc. in Operations Research from Institute Technological of Aeronautics, and PhD in Operations Research from State University of Campinas. His research interest includes Operational Research, Simulation and Logistics. His e-mail address is <fmarins@feg.unesp.br>.

Rafael Florêncio da Silva Costa is a Production Engineering undergraduate student at Federal University of Itajubá. His research interest includes Operations Research and Simulation. His e-mail address is <Rafael.florencio@yahoo.com.br>.

AN APPLICATION FOR WEB-BASED MODELING AND SIMULATION

Yurena García-Hevia Mendizábal[a], I. Castilla[b], R.M. Aguilar[c], R. Muñoz[d]

[a] Engineer at the Computer Science School, La Laguna University, Spain
[b][c][d] Department of Systems Engineering and Automation and Computer Architecture, La Laguna University, Spain

[a]ygarcia@isaatc.ull.es, [b]ivan@isaatc.ull.es, [c]rosi@isaatc.ull.es, [d]rmglez@isaatc.ull.es

ABSTRACT

The disciplines of modeling and simulation are essential to the study of complex systems. The Department of Systems Engineering and Automation and Computer Architecture at the University of La Laguna has been developing a Java simulation library for Discrete Event Systems (SIGHOS). SIGHOS is intended to be used as part of tools that analyze the effects of proposed solutions on real problems. Using this library, however, requires programming skills which limit its effectiveness. In this paper we propose the introduction of a logic layer that separates the low-level simulation mechanisms of the modeling system. We achieve this through the creation of a web service that can run the simulation remotely via the Internet using SIGHOS. We then describe a service-oriented web application that encompasses a modeling and simulation environment through which a user can create models, process them (using the aforementioned web service), and observe the results without any knowledge of programming.

Keywords: web-based solution, simulation, SIGHOS

1. INTRODUCTION

As discussed in previous articles (Aguilar, Castilla, Muñoz, Estévez, Martín and Moreno. 2005; Aguilar, Muñoz, Castilla, Martín, and Piñeiro 2006b) SIGHOS models through the use of a process-based methodology where the elements proceed via stages as determined by the state of the elements themselves. Within each stage, the elements perform different activities which make use of certain resources. These resources have to be available so as to devote part of their time to the execution of said activity.

Until now, the model to be simulated with SIGHOS could be described in one of two ways: by using the functions provided by the Java library, or by employing the XMLGHOS interface (Aguilar, R. Muñoz, Castilla, Martín, and V. Muñoz 2006a) which facilitates its use and automatically generates the models. XMLGHOS describes the system by using two files:

- *XMLModel* defines the resources, the activities, and the paths or *flows* the elements go through.
- *XMLExperiment* specifies the duration and number of simulations to be run, the type of information to be gathered, and the way the elements flowing in the model are generated.

Although the use of XMLGHOS simplifies the modeling, those wishing to use the library have to be familiar with XML. Several new approaches are being addressed to improve the usability of SIGHOS. In this sense, web-based technology would allow the simulation models to be configured using forms which are more familiar to end-users. Moreover, this technology facilitates remote interoperability and accessibility, as well as the integration with a wide range of interfaces available on the market.

2. THE WEB SERVICE

In recent years, web services have become one of the easiest and most efficient ways to access services remotely. The use of web services in simulation remains largely untapped. There are some papers on the topic, but most of them are limited to a specific operating environment (Orsini, Bruzzone and Revetria 2003; Moles, Alonso and Banga 2003; Bruemel, Vinichenko and Novickis 2007), such as assembly lines, transportation services, etc., or use outdated technologies (Taylor 2000) as applets.

Web services may be defined as a set of applications or technologies that are able to interoperate on the web (W3C 2004). These applications or technologies exchange data for the purpose of offering services. The providers offer their services as remote procedures and the users request the service by calling these procedures via the Web. The services provide standard communication mechanisms between the different applications, which interact among themselves to present dynamic information to the user. This interoperability is made possible by the use of open protocols.

The simulation processed by SIGHOS can thus be executed on a high-performance system (computer cluster, supercomputers, dedicated servers…) and accessed via the web service transparently by various clients. These can be written in different programming languages and be running on different hardware and/or software platforms thanks to the use of open standards.

The web service involves adding a new abstraction layer to the SIGHOS simulation architecture, as shown in Figure 1. In this process, a method, which we designate simulate(), is published and made available on the Internet. This process requires a description of the parameters needed to carry out the simulation, namely the *XMLModel* and *XMLExperiment* previously described.

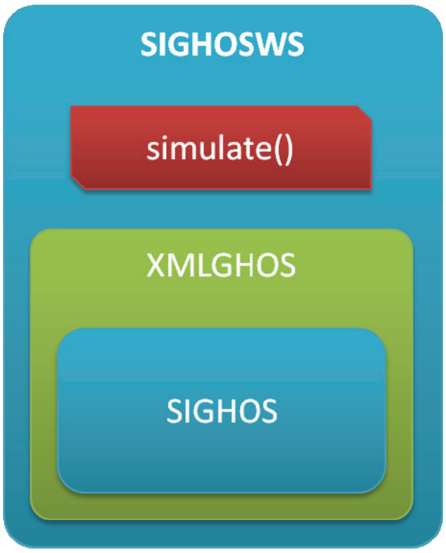


Figure 1: Web service layer (SIGHOSWS) over XMLGHOS and SIGHOS.

In order to facilitate the post-processing of data, the simulate() method returns the results of the simulation in XML format.

3. WEB-BASED MODELING AND SIMULATION ENVIRONMENT

One of the main objectives of our study is to make SIGHOS accessible to those users who are unfamiliar with the currently available interfaces (Java and XML). In this section, we describe the development process of a web application which constitutes a modeling and simulation environment and which allows these users to easily access the powerful characteristics of SIGHOS. This application is known as MOSINET (Modeling and Simulation on the Internet).

3.1. Analysis and design

analysis and design of the application consider the conditions or abilities that must be met by the software system to be developed so as to satisfy the specifications needed to increase the usability of the library.

The demands placed on traditional desktop applications by the user include necessities such as functionality (what the system must do), the storage of information, the ease of use and performance. Within a Web application, in addition to the above, it is also necessary to correctly define user accessibility requirements, its adaptability and scalability, its availability and participation (Cala 2007).

Often, the navigational structure of an application can be comfortable and intuitive for a Web designer who is familiar with the environment, but a user may not be able to identify all the requirements if he lacks an understanding of nodes, links, option menus, search engines, etc. It is important that these requirements be specified in terms:

- that the user knows and understands so as to improve communications between the analyst and the user; and
- that are sufficiently clear and concise so as to be correctly interpreted by the programmers of the web application.

When dealing with a constantly changing environment, the concepts of adaptability and scalability take on added significance. It is important that the system be able to grow without compromising the quality of the services offered, and also that it be able to respond to changes in lower levels of the application.

Since the use of the application will not be subject to strict time schedules and will be accessible to users in different time zones, it must be *available* at all times. This condition can be met through the use of web technologies.

3.2. Application architecture

The application's architecture is structured in three layers, as shown in **Errore. L'origine riferimento non è stata trovata.**. This separation allows for development to take place at several levels (by separating each layer from the whole). In case of a change, it is only necessary to modify the affected level without having to revise the entire code.

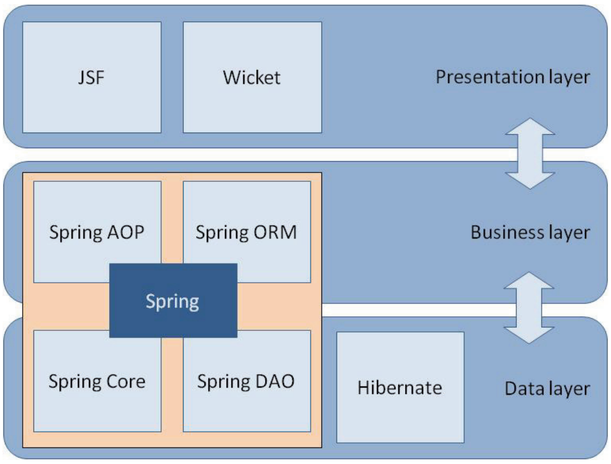


Figure 2: Web application layer structure.

Each layer's functions are as follows:

- *Presentation layer:* the graphical interface which interacts with the user. Used to show information to the user and to record the information provided by the user.
- *Business layer:* used to implement the services to be provided by the application. It receives requests from the presentation layer, processes

them and returns the results. It also communicates with the data layer to store or retrieve information from the database.

- *Data layer:* used to store and manipulate data. It consists of a database and a manager that receives information storage and retrieval requests from the business layer.

The application is built on the Java Spring and Hibernate frameworks. These frameworks simplify the development of the business and data layers by supplying the tools needed for user session management, object handling, framework-API interactions, etc.

3.3. What the user sees

The structure of the application's proposed presentation layer is displayed in **Errore. L'origine riferimento non è stata trovata.**, which shows four strictly classified groups of pages.

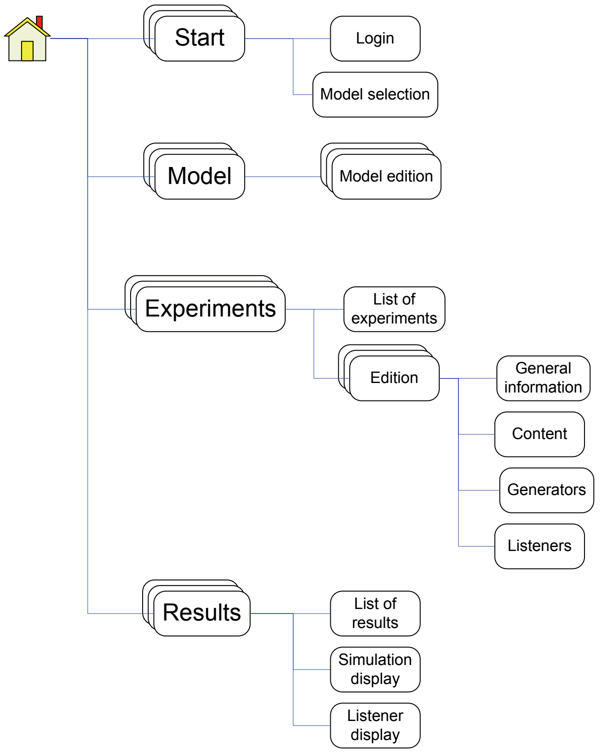


Figure 3: Site map of the web application

- On the start page, the user accesses the system by entering a unique username and password. Once validated, he selects a SIGHOS *XMLModel* with which to work (or creates a new one). A set of *XMLExperiments* is associated with each model. This work environment, in which just one *XMLModel* is used in each session, simplifies the reference checks made between *models* and *experiments*, that is, when reference is made to deleted element or resource types, when invalid identifiers are used, etc.
- On the pages used to define the model, the user can use forms to edit the SIGHOS components graphically. The data input are converted to XML and sent to the web service at the time of simulation. The components described here are the elements, the resource types and their availability, and the activities and work flows through which the elements will be processed.
- All the *XMLExperiments* associated with the model in use can be seen as a list. As with the *model*, the user can graphically edit the information associated with each SIGHOS experiment, including the way the elements are generated, the duration and the results to be recorded during the simulation. This information is stored in XML format and is used along with the model description file to carry out the web service simulations.
- The data from the processed simulations are shown on the results page. The user can select one simulation from a list, and thus see the result outputs returned by the simulator. These results are those which were selected in the experiment, and can be displayed graphically or downloaded on a file (XML, CSV, etc.).

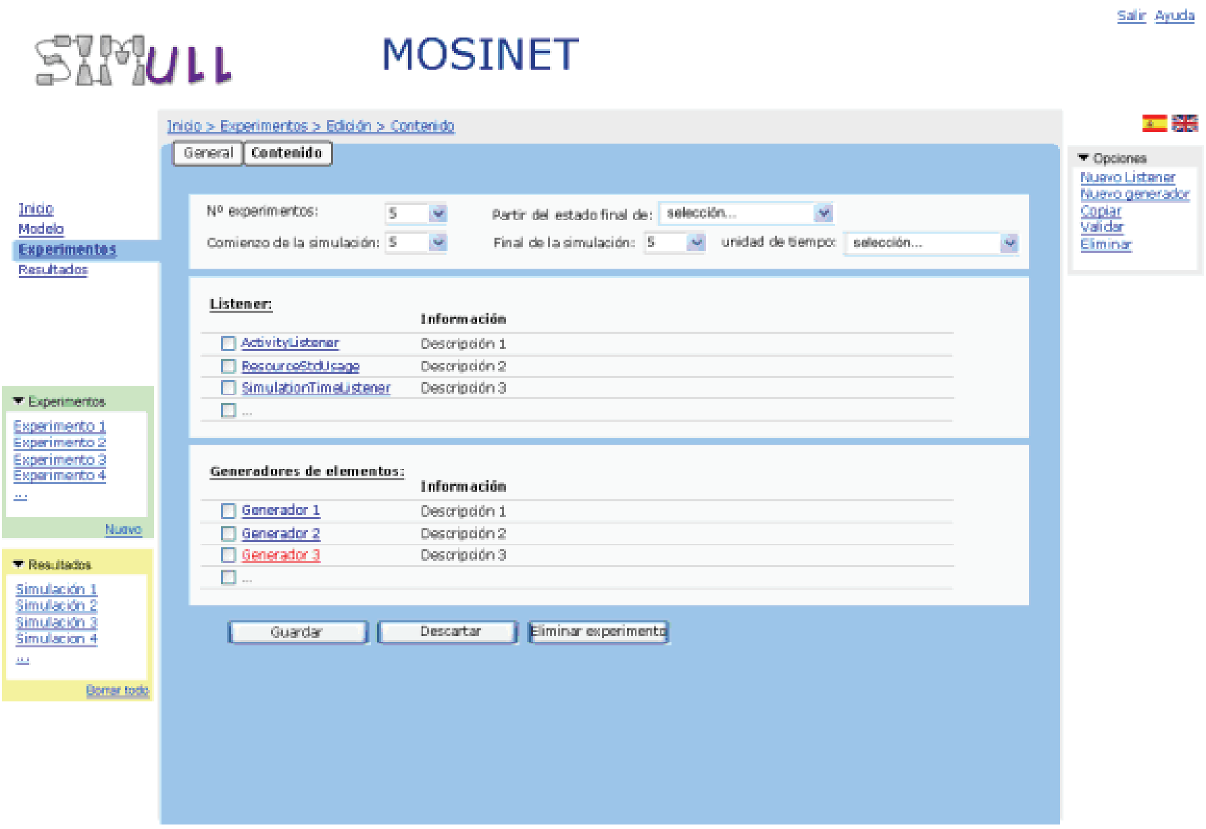
3.4. Use of the application

As described in the previous section, the user works with a single *XMLModel* and its associated *XMLExperiments*. This model can be new, or it can be edited using forms, or imported from an XML document.

Once the *model* is selected, *experiments* can be edited, created, imported and deleted; selected to be simulated; copied to create new experiments from existing ones, etc.

Errore. L'origine riferimento non è stata trovata. shows a prototype of the application. The screen has been divided in four areas:

- *Top pane*: it shows the application's logo, the login controls (you can log in, create an account or log out if you are logged in) and one link to access help (which explains how to use the application).
- *Left menu*: in this zone the user can select from among the four sets of pages previously described (start, models, experiments and results).
- *Main pane*: it shows the content of the page selected on the left menu.
- *Right pane*: there is a list of feature options which can be applied to the contents shown on the main pane (e.g. when viewing the list of available experiments, these options will include the possibility of deleting, copying, editing or exporting an existing experiment to XML, creating or importing a new experiment, etc.).



Nº experimentos: 5
Comienzo de la simulación: 5
Partir del estado final de: selección...
Final de la simulación: 5
unidad de tiempo: selección...

Listener:	Información
☐ ActivityListener	Descripción 1
☐ ResourceStdUsage	Descripción 2
☐ SimulationTimeListener	Descripción 3
☐ ...	

Generadores de elementos:	Información
☐ Generador 1	Descripción 1
☐ Generador 2	Descripción 2
☐ Generador 3	Descripción 3
☐ ...	

Figure 4: Web application prototype

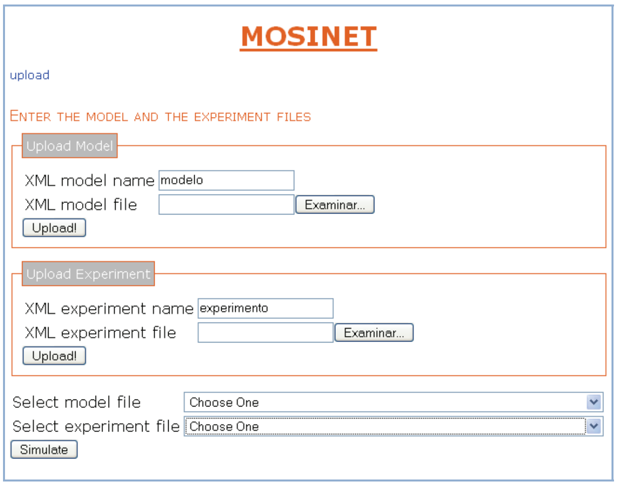
The experiment(s) to be simulated can be selected on the screen where they are displayed. This process then launches simulations which are calls to the published web service's simulate() method. Each selected experiment involves a call to simulate() with the same *XMLModel* and different *XMLExperiment*.

3.5. Results

MOSINET has progressed from the analysis phase into the development phase. To date, the XML interfaces for returning the results of the service have been developed and the web service has been published.

The first version of MOSINET is also available, and some screenshots can be seen in Figure 5 and Figure 6. This version allows users to perform the following operations:

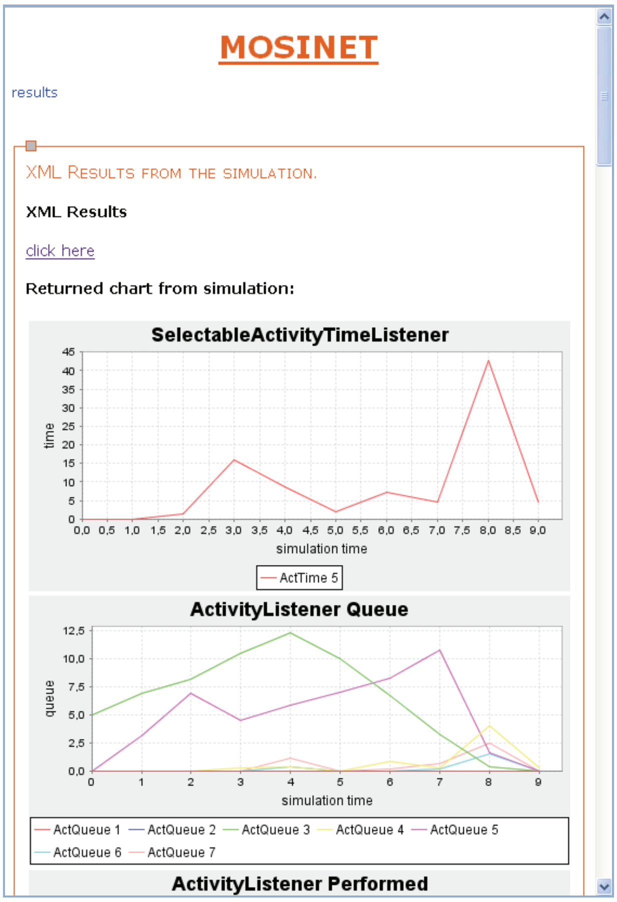
- Upload models and experiments for storage in a database.
- Select a model and experiment to be simulated from among those available. This action uses the published web service.
- Once the simulation is completed, the results obtained are provided to the user in various formats, as shown in Figure 6. The results can also be downloaded in an XML format. The information is also used to generate graphs.



XML model name modelo
XML model file
XML experiment name experimento
XML experiment file
Select model file Choose One
Select experiment file Choose One

Figure 5: Web Interface of the first version of MOSINET

A large portion of the data layer and the initial phase of the business layer has been implemented in the background of this application.



MOSINET

results

XML Results from the simulation.

XML Results

click here

Returned chart from simulation:

Figure 6: Web Interface for the simulation results

4. CONCLUSION

In this paper, we have presented a modeling and simulation framework which simplifies the use of a java simulation library.

A web service has been developed to make SIGHOS and the graphical interface independent, thereby enabling interoperability using standard protocols. This allows us to execute SIGHOS on a computer cluster, and the web interface on another server. Consequently, the capabilities of both two layers of the system are optimized.

In addition, a web interface called MOSINET has been designed to model and simulate with SIGHOS in a more convenient and easier way, hence improving the accessibility to the library for all types of users. The architecture of this interface is also divided into layers so that the data stored for each user and the services offered by the application remain separate.

ACKNOWLEDGMENTS

This work is being supported by a project (reference DPI2006-01803) from the Ministry of Science and Technology with FEDER funds.

Iván Castilla is being supported by an FPU grant (ref. AP2005-2506) from the Spanish Ministry of Education and Science.

REFERENCES

Aguilar R.M., Castilla, I, Muñoz, V, Estévez, J.I., Martín, C.A., Moreno, L., 2005. *Predictive Simulation for Multiagent Resource Distribution: An Application in Hospital Management*. International Mediterranean Modeling Multiconference (EMSS 2005). Marseille- France.

Aguilar, R.M., Muñoz, R., Castilla, I., Martín, C.A., Muñoz, V., 2006. *An interface to integrate workflow diagrams into discrete event simulation*. International Mediterranean Modeling Multiconference. Barcelona- España.

Aguilar, R.M., Muñoz, R., Castilla, I., Martín, C.A., and Piñeiro, J.D., 2006. *Verification and Validation in Discrete Event Simulation: A case study in hospital management*. (EMSS 2006) Barcelona-Spain

Bruemel, E., Vinichenko, S., Novickis, L. 2007. *Model-Based essential logistics principles for creating a web-portal of transport services consumers*. I.J. of SIMULATION Vol. 8 No 1. Available from: http://ducati.doc.ntu.ac.uk/uksim/journal/Vol-8/No-1/Paper-1.pdf [accessed 9 April 2008].

Cala, C., Rodríguez, A.L., and Barroso J.A. 2007. *Reflexiones sobre el Framework de desarrollo del Consejo Superior de Investigaciones Científicas*. Available from:

Moles, C.G., Alonso, A.A., Banga, J.R., 2003, *Desarrollo de Herramientas de Simulación Vía Web Basadas en Ecosimpro*. February 24-25, Madrid (Spain)

Orsini, A., Bruzzone, A.G., Revetria, R. 2003 *Framework Development For Web-Based Simulation Applied to Supply Chain Management*. I.J.of SIMULATION Vol. 4 No. 1. Available from: http://ducati.doc.ntu.ac.uk/uksim/journal/Vol-4/No-1&2/Orsoni.pdf [accessed 9 April 2008].

Taylor, S. 2000. Netmeeting a Tool for Collaborative Simulation Modelling. I. J. of SIMULATION Vol. 1 No 1-2. Available from: http://ducati.doc.ntu.ac.uk/uksim/journal/issue-1/SimonTaylor/SimonTaylor.pdf [accessed 9 April 2008].

W3C, Web Services Architecture. 2004. Available from: http://www.w3.org/TR/ws-arch/ [accessed 15 April 2008].

AUTHORS BIOGRAPHY

YURENA GARCÍA-HEVIA MENDIZÁBAL is a Technical Computing Systems Engineer from the University of La Laguna. She is currently a student of Engineering Computer Science and a research grant holder at the Department of Systems Engineering and Automation at the University of La Laguna. Her fields of interest are the design and implementation of simulation environments for service organizations, modeling and simulation, and business intelligence. Email to <ygarcia@isaatc.ull.es>

IVÁN CASTILLA was born in La Laguna, Tenerife and attended the University of La Laguna, where he studied Engineering Computer Science and obtained his degree in 2004. He is currently working on his PhD with the Department of Systems Engineering and Automation at the same university. His interests include discrete event simulation and business intelligence. Email to <ivan@isaatc.ull.es>

ROSA M. AGUILAR received her MS degree in Computer Science in 1993 from the University of Las Palmas de Gran Canaria and her PhD degree in Computer Science in 1998 from the University of La Laguna. She is an associate professor in the Department of Systems Engineering and Automation at the University of La Laguna, Canary Islands, Spain. Her current research interests are decision making based on discrete event simulation systems and knowledge-based systems, intelligent agents, and intelligent tutorial systems. Email to <rosi@isaatc.ull.es>

ROBERTO MUÑOZ was born in La Laguna, Tenerife and obtained his degree in Engineering Computer Science from the University of La Laguna in 2006. He works as an analyst in Nivaria Innova and collaborates with the Department of Systems Engineering and Automation at the University of La Laguna, where he is working on his PhD. His research interests include the modeling, simulation and optimization of business processes. Email to <rmglez@isaatc.ull.es>

TRANSIENT SIMULATION OF BALANCED BIDDING IN KEYWORD AUCTIONS

Maurizio Naldi[a], **Giuseppe D'Acquisto**[b]

[a]Università di Roma "Tor Vergata", Rome, Italy
[b]Università di Roma "Tor Vergata", Rome, Italy

[a]naldi@disp.uniroma2.it, [b]giuseppe.dacquisto@inwind.it

EXTENDED ABSTRACT

The synchronous version of the Balanced Bidding strategy for keyword auctions is examined through simulation. In the case of a uniform distribution for advertisers' valuation of clicks, advertisers prefer the highest positioned slots. The slot assignment process matches advertisers' expectations with a probability decaying with the number of slots according to a power law and depending negligibly on the number of advertisers.

Keywords: Search engines, Auctions, Keywords, Balanced Bidding

1. INTRODUCTION

Search engines, such as Google or Yahoo!, present users with a set of hyperlinks in response to their queries. In addition to the links deemed relevant to the query by the search engine (often named organic links), a number of sponsored links are presented as well, see Battelle (2005), associated to the query through the keywords specified in it. Advertisers are willing to pay for their ads to appear on the seach engine's response. Such sponsored links are then generally assigned through auctions, and the resulting revenues represent a significant source of income for search engines, see Edelman et alii (2007). On the other hand advertisers are interested inmaking their advertising strategy as efficient as possible. Such auctions have been studied for some time now in the context of game theory, where the players in the game are the auctioneer (the search engine) and the advertisers (see e.g. Varian (2007)). The related studies have been devoted mainly to examining if, and under which conditions, the game exhibits a Nash equilibrium.

Such conditions are typically linked to the assignment and pricing rule on one side and to the advertisers' bidding strategies on the other side. As to the former issue the Generalized Second Price (GSP) rule has reached a wide consensus, which however leaves the field open as to the bidding strategy for the advertiser.

Recently the Balanced Bidding (BB) strategy has been proposed by Cary et alii (2008), where advertisers update their bid at each auction round by exploiting the intelligence gathered in the previous rounds. In this process each advertiser identifies at each round his optimal slot as that maximing his utility. In Cary et alii (2008) the convergence to a Nash equilibrium has been studied, but the optimal slot determination process has not been explored in detail, though its relationship with the subsequent slot assignment is central to the advertiser's satisfaction. In this paper we analyse the characteristics of preferences for slots observed for advertisers as resulting from the repetition of such keyword auctions, when advertisers follow the BB strategy. In particular we examine the way advertisers distribute their preferences among the slots on auction, and how the auction's results match their expectations as to the assigned slot.

2. KEYWORDS AUCTIONS

Search engines act in response to users' queries for websites containing the information of interest. In such queries the information of interest is synthetically expressed as a string of keywords, possibly connected through Boolean operators. For the example, in response to the query "*sea* AND *winds* NOT *ice*" the search engine will return pointers to all the documents containing the first two terms but not the third one. The hyperlinks returned by the search engine are typically named organic links. The search engine can add to this list (and show e.g. on the right-hand side of the screen) a number of sponsored links. The available positions for sponsored links are named slots. Such links are provided by advertisers, who are willing to pay to have their ad appear on the screen in relation to a query containing a specific keyword. Hence for any query there are a number of potential fillers of the screen space devoted to sponsored links. It is assumed that the advertisers choose to run for keywords that are actually related to their product. The payment rules may be freely defined in the contract relationship between the search engine manager and the advertiser, but the most established agreement follows the pay-per-click model, where the advertiser pays a pre-determined amount of money each time the user actually clicks on the ad. Since the number of slots is generally smaller than the number of interested advertisers (i.e. advertisers who have opted to run for a keyword appearing in the query), slots represent a scarce resource and a natural way to assign them to the advertisers is through auctions, namely keywords auctions. Hence advertisers declare how much they are willing to pay for a click, and an auction is run for the slots among the advertisers whose keywords match the query. We have therefore a number of slots $S \in Z^+$ and a larger number of advertisers $A \in Z^+$, with $A < S$. Actually, a new auction is run every time a query is submitted, among the advertisers submitting bids for keywords matching the query. For any given keyword we have then a sequence of repeated auctions. As will be seen in Section 4, the repetition of auctions allows advertisers to update their bids by taking into account their past observations of other bidders' behaviour and of the output of previous auction runs.

In order to make the assignment process as effective as possible the auctioneer has to carefully design the auctioning rules, which boils down to choosing: a) the assignment rule (i.e. the way advertisers are assigned the slots); b) the price setting rule (i.e. the price an advertiser has to pay when the user clicks on its ad).

As to the first issue this is unanimously solved by using a straightforward ordering of slots and advertisers. Slots are indexed progressively by their vertical position on the screen (the slot appearing on top of the screen is assigned index 1 by convention; the slot appearing on the bottom of the ad-devoted space has index *S*) and evaluated by their click-through rate. The click-through rate θ_i of slot *i* is the probability that the user clicks on that slot. Its estimate can be obtained by dividing the number of users who clicked on an ad on a web page by the number of times the ad was delivered (impressions), see Sherman and Deighton (2001). It is generally accepted that the click-though rate is a declining function of the slot's position, i.e. $\theta_i > \theta_{i+1}$, when $i = 1,\ldots,S-1$; a statistical study reported in Brooks (2004) supports this assumption. Hence, top-positioned slots are more valuable than bottom-positioned slots. As to the precise shape of the click-through rate decaying function, we consider a Zipf distribution for the probability that the user clicks on a given slot. Namely the probability that the user clicks on the slot *j* is

$$\theta_j \propto \frac{1}{j^\alpha}, \tag{1}$$

where $\alpha \in R^+$ is the Zipf parameter.

For convenience (with no consequence on the following results) we adopt the normalizing condition $\sum_{j=1}^{S} \theta_j = 1$, so that we are actually considering the probability of clicking on a specific slot conditioned to the user clicking on a slot (or, alternatively, the user clicks on a slot with probability 1). Though in this paper we implicitly consider the click-through rate being a function of the slot's position only, other authors have considered the more general case of click-through rates being function of advertisers as well , see Feldman and Muthukrishnan (2008) and Aggarwal et alii (2006). Advertisers are likewise ordered in a decreasing function by the value of their bid.

If we now denote by b_i the bid submitted by the *i*-th advertiser, and then by $b_{(j)}$ the *j*-th highest bid, the assignment rule states that the *k*-th slot is assigned to the advertiser submitting the bid $b_{(k)}$. For convenience we introduce the function $\Pi(k)$ returning the index of the advertiser who's assigned the *k*-th slot.

Setting the price is a less straightforward matter. A well-known mechanism is the truthful Vickrey-Clarke-Groves rule, which would lead each participating advertiser to bid its true valuation, see Clarke (1971) and Groves (1973). However, search engines do not adopt the VCG mechanism in practice, but rather the Generalized Second Price (GSP) rule, which is described in detail in Section 3.

3. GENERALIZED SECOND PRICE MECHANISM

In this Section we review the basic characteristics of GSP as a price setting mechanism. In GSP the natural assignment rule is maintained whereby the advertiser submitting the *k*-th highest bid $b_{(k)}$ is assigned the *k*-th slot. However the price he pays is equal to the next lower bid, i.e. $p_{(k)} = b_{(k+1)}$. Advertisers who are not assigned a slot pay nothing. The most important decision advertisers have to take is then to choose their bids. As a reference they have their own private valuation of clicks: in the simplest scenario the *i*-th advertiser values a click worth v_i (i.e. the click value doesn't depend on the slot position itself and doesn't vary as the auction is repeated). In general any bid of the generic *i*-th advertiser will satisfy the inequality $b_i \leq v_i$. The expected utility of the advertiser receiving the the *k*-th slot is $\theta_k\left(v_{\Pi(k)} - b_{(k)}\right)$. The most important property of the VCG mechanism is that it induces the advertiser to declare its private valuation, so that $b_i = v_i$ (truthfulness property). On the contrary, GSP is not truthful, hence advertisers' bids are limited by the above inequality only. If we consider the static game associated to GSP-driven auction, a Nash equilibrium has been shown to exist, see Varian (2007). However, in the dynamic version resulting from the repetition of the auction, bidders can update their bid at each new issue of the auction by taking advantage of the knowledge they have gained from the past auction occurrences (the bids submitted by all the other bidders).

4. BALANCED BIDDING STRATEGY

If the advertiser dosn't submit a truthful bid, he has full freedom to choose for his bid any value satisfying the inequality recalled in Section 3. Cary et alii have proposed the Balanced Bidding (BB) strategy, see Cary et alii (2008), under which the advertiser chooses his next bid b so as to be indifferent between successfully winning the targeted slot k at the price p_k at which it was awarded at the previous auction run, or winning the slightly more desirable slot *k-1* at price b. In this context a set of A advertisers, who compete for S slots and have their private valuations $\{v_1, v_2, \ldots, v_S\}$ for a click, are assigned the S slots according to the GSP rule. The resulting BB strategy leads the winning i-th advertiser to:

1. targeting the slot k_i^* that maximizes its utility (optimal slot), i.e. $k_i^* = \arg\max_k \{\theta_k (v_i - p_k)\}$;
2. setting its next bid b according to the expression $b_i = v_i - \frac{\theta_{k_i}}{\theta_{k_{i-1}}}\left[v_i - p_{k_i^*}\right]$.

Losing advertisers' bids instead equal their valuations.

For the asynchronous version of BB (advertisers update their bids one at a time) Cary et alii have proved that the dynamic system where all bidders play this strategy converge to a unique fixed point, which is also the Nash equilibrium of the static game, see Cary et alii (2008). However, the convergence time depends on the number of bidders and may take some hundreds of auction runs.

In this paper we focus on the transient behaviour of the auction in the synchronous case, i.e. under the hypothesis that each bidder plays the Balanced Bidding strategy at each round.

5. MEASURES OF AUCTION'S SUCCESS

In order to evaluate the characteristics of the auction a number of metrics can be considered. In Naldi and D'Acquisto (2008) some have been proposed to reflect the interest of the bidders as well as that of the auctioneer. Since in the Balanced Bidding strategy bidders submit their bids after a process identifying the optimal slot, it is natural to consider their bids as referred to a specific slot (though the slot is not mentioned explicitly in the submission process). If we consider auctions as a matching process between the objects on auctions (the slots) and the bids, a measure of success is naturally given by the probability that a slot is assigned to the bidder for which that slot is optimal. We name such probability the slot matching probability. If we indicate by Y_i the optimal slot of the bidder who is assigned the slot i, the formal representation of such probability is then $\omega(i) = P[Y_i = i]$ for a generic i, or, if we lose the details of the specific slot, the average

$$\Omega = \frac{1}{S} \sum_{i=1}^{S} \omega(i). \qquad (2)$$

If all the users were to designate the same slot as their optimal one, the slot matching probability would be equal to the inverse of the number of slots. Though the advertisers' preferences are not so unanimous, they are far from being uniformly distributed, so that the same slot may be regarded as optimal by multiple advertisers while some other is not even for a single advertiser. Another issue of interest is then the distribution of such preferences, i.e. the values of $\psi(i,j) = P[Y_i = j]$ when $j = 1,2,\ldots,S$ and $i = 1,2,\ldots,A$. Here, similarly, we can consider just the average over the set Λ of winning advertisers

$$\Psi(j) = \frac{1}{S} \sum_{i \in \Lambda} \psi(i,j). \qquad (3)$$

In this paper we focus on these two measures: the slot matching probability and the distribution of preferences. We resort to MonteCarlo simulation, by running N_{sim} times a simulation cycle consisting of T repetitions of the auction for S slots. In Cary et alii (2008), where the Balanced Bidding strategy has been proposed, the asynchronous version of that strategy is considered, where a single advertiser (randomly chosen) is given the chance to update his bid at a time. This leads to a quite slow convergence towards the VCG results: roughly 200 rounds are needed for the players' payoffs to converge. Here we instead take the much more realistic assumption that all advertisers actually update their bid at each round. Such asumption is expected to reduce considerably the convergence time. Our study will therefore concentrate on the transient behaviour of the auction, i.e. that pertaining to the first batch of repetitions.

For evaluation purposes we have to set some working hypotheses, in particular concerning the advertisers' valuations and the click-through rate (for which we assume the Zipf distribution). A relevant role in the auction's outcome depends on the distribution of bidders' valuations (which remains unchanged during the subsequent repetitions of the auction). In Naldi and D'Acquisto (2008) a number of distributions were considered: uniform, triangular, Gaussian, exponential, and Pareto. They can be roughly divided into two sets, respectively comprising those showing a large dispersion of bids and those where bids cluster around a common value. In the present early study we consider just the uniform distribution as a representative of the first category. The valuations are then represented by the i.i.d. random variables $V_1, \ldots, V_A$ following a standard uniform distribution, so that the coefficient of variation (standard deviation-to-expected value ratio) is $1/\sqrt{3}$.

6. DISTRIBUTION OF PREFERENCES

As stated in the previous section we first consider the distribution of preferences, i.e. the probability that a given slot position is deemed as optimal by the advertisers. Since the definition of optimality is based on the evaluation of the utility, which is positive only if the advertiser's private valuation is larger than the current price for that slot, not all advertisers declare an optimal slot. Hence the distribution is evaluated for the restricted group of addvertisers who do exhibit an optimal slot. The evaluation is conducted by simulation. We consider N_{sim} MonteCarlo simulation runs, each consisting of T repetitions of the auction for S slots. Within each run any auction is conducted according to the GSP mechanism, where each advertiser adopts the Balanced Bidding strategy described in Section 4. At the end of the full set of MonteCarlo simulation runs we can estimate the distribution of preferences $\Psi(j)$ as a function of the slot's position $j = 1,\ldots,S$. The estimator's expression is

$$\widehat{\Psi}(j)=\frac{1}{N_{sim}\cdot T\cdot S}\sum_{i=1}^{N_{sim}}\sum_{t=1}^{T}\sum_{k=1}^{S} I_{\left[x_k^{(t,i)}=j\right]}, \qquad (4)$$

where $I_{[\bullet]}$ is the indicator function (equal to 1 if its logical argument is satisfied and zero otherwise) and $x_k^{(t,i)}$ is the optimal slot (as resulting after the auction repetition *t*) of the bidder who is assigned the slot *k* at the repetition *t* in the *i*-th simulation run. Here we report some results obtained under the following conditions:

- Number of simulations runs $N_{sim} = 10000$
- Number of slots $S = 5,10$
- Number of advertisers $A = S + 1$
- Number of repetitions $T = 100$
- Uniform distribution of advertisers' private valuations
- Zipf distribution for the probability of users clicking on a given slot, with the Zipf parameter $\alpha = 0.5,1,2$

We briefly review these assumptions. The number of simulation runs is large enough to allow for an excellent accuracy for the values at hand. The relative standard error of this crude MonteCarlo estimator $\widehat{\Psi}(j)$ is in fact

$$\varepsilon=\sqrt{\frac{1-\Psi(j)}{N_{sim}\cdot T\cdot S\cdot\Psi(j)}}\,. \qquad (5)$$

As to the number of slots, the size of the screen space available for sponsored links coupled with the visibility requirements can hardly allow for more than 10 such links. When we come to the assumption on the number of advertisers, actually we could imagine a number much larger than the number of slots. However, by choosing *A* as the minimum integer larger than the number of slots, we reduce to a minimum the computational load, while obtaining a final result quite accurate also for larger values of *A* (as briefly shown later). In Figure 1 and Figure 2 we draw the distribution of preferences when the number of slots is 5 and 10 respectively. As expected the figures for the 10 slots case are generally lower than the 5 slots case, since preferences distribute among a larger number of potentially optimal slots. In both cases we can however note that the highest slot is the most preferred one, but at the same time the preference doesn't decay monotonically with the slot position. In fact, in some cases we see a slight upsurge of the preference probability for the lowest slot. Counter-intuitive is also the impact of the Zipf parameter: Highly skewed click-through rate distributions (i.e. having larger values of the Zipf parameter) produce more balanced distributions of preferences.

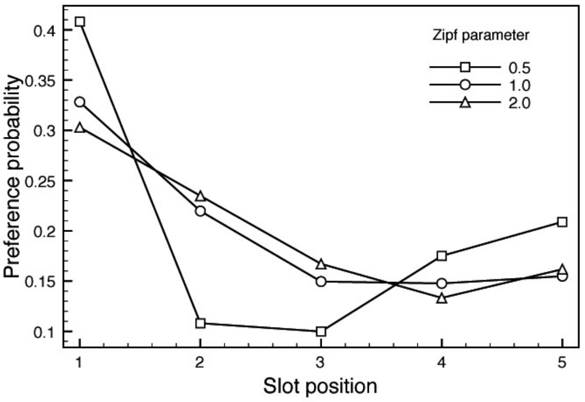


Figure 1: Probability of Advertisers' Preferences with 5 Slots

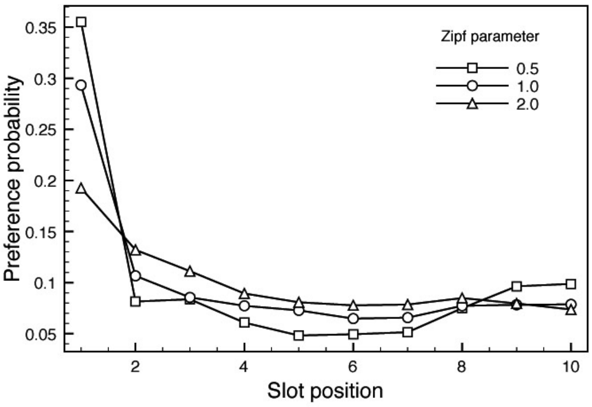


Figure 2: Probability of Advertisers' Preferences with 10 Slots

Finally, we come to the assumptions on the number of advertisers. The results so far shown, obtained for $A = S + 1$, keep valid as long as the preference distribution depends negligibly on the number of advertisers. In Figure 3 we show the same distribution, obtained for $A = 6$ and $A = 50$, where multiplying the number of advertisers tenfold produces a very limited variation on the estimated preference probability, comparable to the accuracy of the simulation method. Hence the results obtained above can be deemed accurate enough for larger values of the number of advertisers as well.

7. SLOT MATCHING PROBABILITY

We now turn to the slot matching probability, which measures how well the slot assignment satisfies the advertisers' expectations, and can therefore be considered as a measure of success of the auction.

After running N_{sim} MonteCarlo simulation runs, each consisting of T repetitions of the auction for k slots, the slot matching probability is estimated as

$$\widehat{\Omega} = \frac{1}{N_{sim} \cdot T \cdot S} \sum_{i=1}^{N_{sim}} \sum_{t=1}^{T} \sum_{k=1}^{S} I_{\left[x_k^{(t,i)}=k\right]}, \qquad (6)$$

where $I_{[\bullet]}$ is the indicator function (equal to 1 if its logical argument is satisfied and zero otherwise) and $x_k^{(t,i)}$ is the optimal slot (as resulting after the auction repetition t) of the bidder who is assigned the slot k at the repetition t in the i-th simulation run.

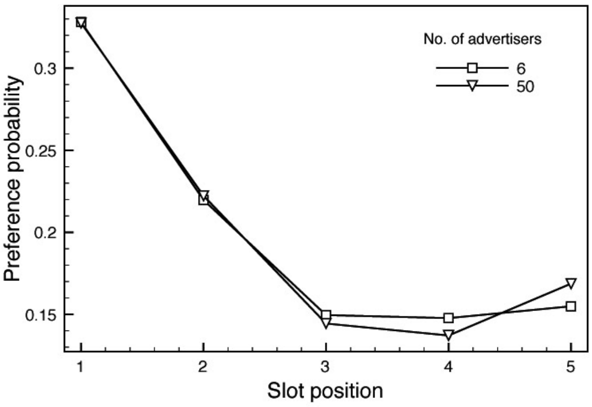


Figure 3: Impact of the Number of Advertisers on Preference Distribution

We report here the results obtained under the following conditions:

- Number of simulations runs $N_{sim} = 10000$
- Number of slots $S = 2{,}10$
- Number of advertisers $A = S + 1$
- Number of repetitions $T = 100$
- Uniform distribution of advertisers' private valuations
- Zipf distribution for the probability of users clicking on a given slot, with the Zipf parameter $\alpha = 0.5{,}1{,}2$

In Figure 4 the slot matching probability appears as a fast decaying function of the number of slots. Matching appears to be rarer for lower values of the Zipf parameter (i.e. as the click rate becomes more uniform over the set of slots).

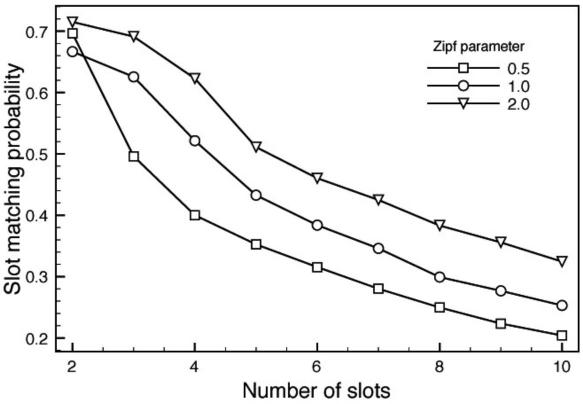


Figure 4: Slot Matching Probability

A tentative fitting can be considered by using the power law model

$$\Omega = \frac{\gamma}{S^{\beta}}, \qquad (7)$$

where γ and β are two constants. For the three values of the Zipf parameter considered in this paper we obtain by regression the values reported in Table 1 along with the resulting R^2 goodness of fit index. We note that, though the fit is generally good, the power law exponent is not a monotonic function of the Zipf parameter.

Table 1: Fitted Power Law Parameters

α	γ	β	R^2
0.5	1.1147	0.7237	0.9935
1	1.4683	0.7574	0.9969
2	1.4417	0.6378	0.9915

In order to analyse the transient dynamics we have also evaluated the slot matching probability through a sliding window of width equal to 20 auction rounds over a total length of $T = 500$ repetitions. The quantity of interest is now a function of the ending time T_{end} of the sliding window

$$\widehat{\Omega}(T_{end}) = \frac{1}{N_{sim} \cdot 20 \cdot k} \sum_{i=1}^{N_{sim}} \sum_{t=T_{end}-20}^{T_{end}} \sum_{j=1}^{k} I_{\left[y_j^{(t,i)}=j\right]}, \qquad (8)$$

where $T_{end} = 21, \ldots, 500$.

A rough analysis of the length of the transient can be obtained by visual inspection of the resulting slot matching probability. In Figure 5 (obtained for $S = 5$ slots and for the Zipf parameter $\alpha = 1$) the curve stabilizes well before the first 40 rounds, much earlier than the convergence time reported in Cary et alii (2004) for the asynchronous version of the auction.

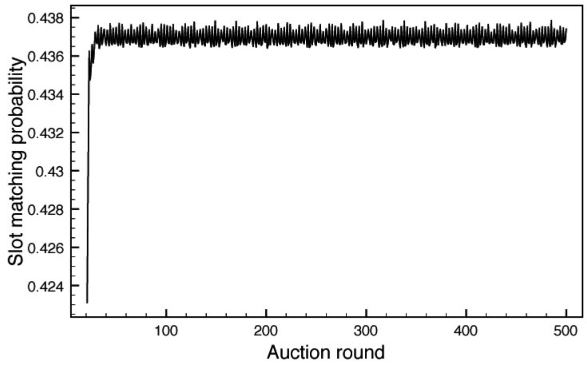


Figure 5: Transient in Slot Matching Probability

8. CONCLUSIONS

In this paper we have examined some characteristics of the Balanced Bidding strategy applied to repeated auctions for keywords. We have considered the synchronous version of that strategy (all the bidders update their bids at each auction round). As to the performance indices we have focussed on the probability that a slot is assigned to the advertiser for which it is optimal, which can be considered as a measure of success of the advertiser's bidding strategy. The slot matching probability depends negligibly on the number of bidders, while decays fast with the number of slots on sale. For that relationship we provide a tentative power law fitting, where however the power law exponent doesn't vary monotonically with the Zipf parameter. We have also examined the way bidders' preferences distribute among the available slots. The highest slots are the most preferred ones, but the lowest slots may exhibit a preference upsurge. Such phenomenon, which makes the preference distribution non monotonic, is more prominent the lower the Zipf parameter.

REFERENCES

Aggarwal, G. , Goel, A., Motwani, R., 2006. Truthful auctions for pricing search keywords. *EC '06: Proceedings of the 7th ACM conference on Electronic commerce*, 1–7, 11-15 June 2006, New York, NY, USA.

Battelle, J., 2005. *The Search*. Portfolio Hardcover.

Brooks, N., 2004. The Atlas Rank Report: HowSearch Engine Rank Impacts Traffic. *Technical report, Atlas Institute.*

Cary, M. , Das, A., Edelman, B., Giotis, I., Heimerl, K., Karlin, A.R., Mathieu, C., Schwarz, M., 2008. On Best-Response Bidding in GSP Auctions. *NBER Working Paper No. W13788.*

Clarke, E.H., 1971. Multipart pricing of public goods. *Public Choice*, 11(1):17–33.

Edelman, B. , Ostrovsky, M. , Schwarz, M.,2007. Internet Advertising and the Generalized Second-Price Auction: Selling Billions of Dollars Worth of Keywords. *The American Economic Review*, 97(1):242–259.

Feldman, J. , Muthukrishnan, S., 2008. Algorithmic methods for sponsored search advertising. *arXiv Preprint 0805.1759.*

Groves, T., 1973. Incentives in Teams. *Econometrica*, 41(4):617–31.

Naldi, M., D'Acquisto, G., 2008. Performance of the Vickrey auction for digital goods under various bid distributions. *Performance Evaluation*, 65:10–31.

Sherman, L. , Deighton, J., 2001. Banner advertising: Measuring effectiveness and optimizing placement. *Journal of Interactive Marketing*, 15(2):60–64.

Varian, H., 2007. Position auctions. *International Journal of Industrial Organization*, 25:1163–1178.

AUTHORS BIOGRAPHY

Maurizio Naldi graduated cum laude in 1988 in Electronic Engineering at the University of Palermo and then received his Ph.D. in Telecommunications Engineering from the University of Rome "Tor Vergata". After graduation he pursued an industrial career, first at Selenia as a radar designer (1989–1991), and then in the Network Planning Departments of Italcable (1991–1994), Telecom Italia (1995–1998), and WIND (1998–2000) where he was appointed Head, Traffic Forecasting & Network Cost Evaluation Group. In the 1992–2000 period he was active in the standardization bodies (ETSI and ITU), in particular as Associate Rapporteur for Broadband Traffic Measurements and Models at ITU Study Group 2. Since 2000 he is with the University of Rome at Tor Vergata, where he is now Aggregate Professor.

Giuseppe D'Acquisto graduated cum laude in 1995 in Electronic Engineering and received the PhD in Telecommunications in 1999 from the University of Palermo-Italy, with a thesis on rare event simulation. After PhD he started a consulting career, working for Telco Operators and ICT companies, with a focus on market forecasting, traffic engineering and cost accounting. He collaborates with the Universities of Rome Tor Vergata and Palermo in researches in the area of simulations and stochastic optimization. He is the author of more than 20 publicatons on these topics.

A MARKOV PROCESS FOR REFLECTIVE PETRI NETS

Lorenzo Capra

D.I.Co, Università degli Studi di Milano, Italy

capra@dico.unimi.it

ABSTRACT

The design of dynamic (adaptable) discrete-event systems calls for adequate modeling formalisms and tools able to manage possible changes occurring during system's lifecycle. A common approach is to pollute design with details that do not regard the current system behavior, rather its evolution. That hampers analysis, reuse and maintenance in general. A Petri net based reflective model (based on classical PN) was recently proposed to support dynamic discrete-event system's design, and was applied to dynamic workflow's management. Behind there is the idea that keeping functional aspects separated from evolutionary ones, and applying evolution to the (current) system only when necessary, results in a simple formal model on which the ability of verifying properties typical of Petri nets is preserved. On the perspective of implementing in the short time a discrete-event simulation engine, reflective Petri nets are provided in this paper with a timed state-transition graph semantics, defined in terms of a Markov process.

Keywords: stochastic Petri nets, dynamic systems, evolution, state-transition graph, symbolic techniques.

1. INTRODUCTION

Most existing discrete-event systems are subject to evolution during their lifecycle. Think e.g. of mobile ad-hoc networks, adaptable software, business processes, and so on. Designing dynamic/adaptable discrete-event systems calls for adequate modeling formalisms and tools. Unfortunately, the known well-established formalisms for discrete-event systems, such as classical Petri nets, lack features for naturally expressing possible run-time changes to system's structure. An approach commonly followed consists of polluting system's functional aspects with details concerning evolution. That practice hampers system analysis, reuse and maintenance

A Petri net-based reflective model (Capra and Cazzola 2007) was recently proposed to support dynamic discrete-event system's design, and was successfully applied to specify dynamic workflows (Capra 2008). This approach is based on a reflective layout formed by two logical levels. The achieved clean separation between functional and evolutionary concerns results in a simple formal model for systems exhibiting a high dynamism, in which the analysis capabilities of traditional Petri nets should be preserved.

On the perspective of implementing in the short time a discrete-event simulation engine, the Petri net-based reflective model is provided in this paper with a timed semantics, defined in terms of a Markov process. A crucial issue that is handled is about recognizing possible equivalent base-level's evolutions during simulation. That major topic is managed by exploiting the symbolic state (marking) definition that the particular high-level Petri net flavor used at the meta-level (Stochastic Well formed colored Nets, or SWN) is provided with.

The paper balance is as follows: in section 2 a snapshot of reflective Petri nets is given; in section 3 the main features of the employed Petri net classes are presented; in section 4 a stochastic state-transition graph semantics for reflective Petri nets is defined. Finally section 5 is about work-in-progress. Assuming the readers have some basic knowledge about Petri nets, a semi-formal presentation is adopted, where unessential notions are skipped and simple running examples are used.

2. REFLECTIVE PETRI NETS

The *reflective Petri net* approach permits developers to model a discrete-event system and separately its possible evolutions, and to dynamically adapt system's model when evolution occurs.

The approach is based on a reflective architecture structured in two logical layers (figure 1). The first one, called *base-level*, is an *ordinary Petri net* (a P/T net with priorities and inhibitor arcs) representing the system prone to evolve (*base-level PN*); while the second layer, called *meta-level*, consists of a *high-level Petri net* (a colored Petri net) representing the evolutionary strategies (the meta-program, following the reflection parlance) that drive the evolution of the base-level when certain conditions/events occur.

Meta-level computations in fact operate on a representative of the base-level, called (base-level) *reification*. The reification is defined as a (high-level Petri net) marking, whose a portion, encoding the base-level PN current state (marking), is updated every time the base level Petri net enters a new state. The reification is used by the meta-program to observe (*introspection*) and manipulate (*intercession*) the base-level PN. Any change to the reification is reflected on the base-level PN at the end of a meta-computation (*shift-down*).

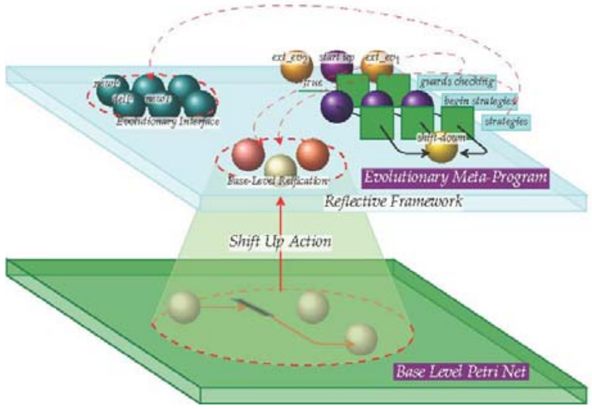


Figure 1: Snapshot of the reflective PN model.

The meta-program is implicitly activated (*shift-up*), then a suitable strategy is put into action, under two conditions: i) either when it is triggered by an external event, or ii) when the base-level enters a given state. The *reflective framework*, a high-level Petri net component as well, is responsible for really carrying out the base-level evolution in a transparent way. Intercession on the base-level PN is carried out in terms of a minimal but complete set of basic operations (called the *evolutionary interface*): addition/removal of places, transitions, arcs - change of transition priorities (base-level's structure change), free moving tokens overall the base-level PN places (base-level's state change).

If one such operation reveals inconsistent, the meta-program is restarted and any changes caused in the meanwhile to the base-level reification are discarded. In other words, evolutionary strategies have a transactional semantics. After a strategy's succeeding execution, changes are reflected down to the base-level Petri net.

Developers have been provided with a tiny ad-hoc language, inspired to Hoare's CSP, that allows anybody to specify his own strategy in a simple way, without any skills in high-level Petri net modeling being required. An automatic translation to a corresponding high-level Petri net is done. Several strategies could be candidate for execution at a given instant: different policies might be adopted in that case to select one, ranging from a deterministic choice to a static assignment of priorities.

According to the reflective paradigm, the base-level runs irrespective of the meta-program, being even not aware of its existence. That raises consistency issues, that are faced by determining, for any strategies, local influence areas on the base-level that are temporarily locked by the meta-level while strategies are being executed.

The interaction between base- and meta- levels, and between meta-level entities, is formalized in (Capra and Cazzola 2007). Let us only outline some essential points:

- The structure of the reflective framework is fixed, while the evolutionary strategies are coupled to the base-level PN, so they vary from time to time. More precisely, the meta-program's model is built according to a predefined pattern, whose the strategies represent the variable component.
- The reflective framework and the meta-program are separated (high-level Petri net) components, sharing two disjoint sets of boundary places denoted hereafter *reification-set* and *evolutionary interface*, respectively. Their composition through a simple place superposition gives rise to the meta-model, called hereafter *meta-level PN*.
- The reification-set is formed by the following colored places: $\{reif_N, reif_M, reif_A, reif_\Pi\}$. The corresponding color domains will be specified later. A well-defined marking of this set of places, hereafter simply denoted *reification*, encodes the structure (including nodes, i.e., places and transitions, arc connections and transition priorities), and the current marking, of the base-level PN. What is most important, there is a one-to-one correspondence, formalized by a bijection, between reifications and P/T nets.
- The initial reification, from time to time, refers to the base-level Petri net modeling the initial system configuration.
- The shift-up, and the reification update following any base-level change of state, are implemented in transparent way at net level, by suitably connecting any base-level PN transition to place $reif_M$, that holds the reification of base-level PN's current marking. The resulting whole model will be hereafter denoted *base-meta PN*.
- The shift-down, i.e., the reflection of changes performed by the meta-program, is modeled by a homonym highest-priority transition of the meta-level PN; it is a kind of meta-transition, that adheres to the usual firing rule as concerns the meta-level PN, further, it makes the (current) base-level PN to be replaced by the P/T net encoded by the reification.

The fixed part of the reflective architecture (the reflective framework) is used to put evolution into practice for any kind of system being modeled. It is responsible for the reflective behavior of the architecture, hiding the work of the evolutionary component to the base-level PN. This approach permits a clean separation between evolution and evolving system, and prevents the base-level PN from being polluted by details related to evolution.

3. SWN BASICS

For (performance) analysis purposes, we decided to use for the base- and meta- levels Generalized Stochastic

Petri nets (GSPN) (Marsan, Balbo, and Conte 1983) and their high-level counterpart, i.e., Stochastic Well formed nets (SWN) (Chiola, Dutheillet, Franceschinis, and Haddad 1993), respectively. This choice has revealed convenient for two reasons: first, the timing semantics of reflective Petri nets is in large part inherited from GSPN (SWN) timing semantics; secondly, the symbolic state representation the SWN formalism is provided with can be exploited to efficiently handle the issues related to recognizing equivalences during model's evolution, as explained in section 4.1.

Let us just recall the basic aspects about SWN (GSPN) timed semantics.

In GSPN (SWN) A priority level is associated to each transition: priority level 0 is reserved for *timed transitions*, while greater priority levels are for *immediate transitions*, which fire in zero time. A *rate*, characterizing an exponential firing delay, is associated to each timed transition, while a *weight* is associated to each immediate transition, to probabilistically solve conflicts between enabled immediate transitions with equal priority.

As a result of this time representation, the *reduced reachability graph* of a GSPN (SWN), i.e., the state-transition graph obtained by suitably removing those markings (called *vanishing*) enabling some immediate transitions, is isomorphic to a Continuous Time Markov Chain (CTMC). Because of the structured syntax of SWN color annotations, behavioral symmetries can be automatically discovered and exploited to build an aggregate state space (called *symbolic reachability graph* or SRG) and a corresponding *lumped CTMC* from a SWN model, according to the *strong lumpability* notion.

As concerns the reflective PN model, while the evolutionary framework, that should be considered as a transparent layer, is formed by immediate transitions only, the evolutionary strategies and, of course, the base-level PN, may also contain timed transitions representing time consuming activities (think e.g. of a network reconfiguration).

3.1. Color Annotations and Symbolic Markings

In Colored Petri nets places, as well as transitions, are associated to *color domains*, i.e., tokens in places have an identifier (color), similarly transitions are parameterized, so that different *color instances* of a given transition can be considered. A *marking m* maps each place *p* to a multiset on the corresponding color domain, $\mathcal{C}(p)$. Any arc connecting *p* to a transition *t* is labeled by a function mapping any element of $\mathcal{C}(t)$ (i.e., any color instance of *t*) to a multiset on $\mathcal{C}(p)$.

The peculiar and interesting feature of the SWN formalism is the ability of capturing system's symmetries thanks to the structured syntax of color annotations. Efficient analysis/simulation algorithms can be applied that exploit such symmetries. These algorithms rely upon the notion of *symbolic marking* (SM).

SWN color domains are defined as Cartesian products of *basic color classes* C_i, that may be in turn partitioned into *static subclasses* $C_{i,k}$. A SM provides a syntactical equivalence relation on ordinary colored markings: two markings belong to the same SM if and only if they can be obtained from one another by means of permutations on color classes that preserve static subclasses. A SM is formally expressed in terms of dynamic subclasses.

3.1.1. SM formal definition.

The definition of a SM (denoted *sm*) comprises two parts specifying the so called dynamic subclasses and the distribution of colored symbolic tokens (tuples built of dynamic subclasses) over the net places, respectively.

Dynamic subclasses define a parametric partition of color classes preserving static subclasses: let D_i and s_i denote the set of dynamic subclass of C_i (in *sm*), and the number of static subclasses of C_i (if C_i is not split then $s_i = 1$). The j-th dynamic subclass of C_i, $Z_j^i \in D_i$, refers to a static subclass, denoted $d(Z_j^i)$, $1 \leq d(Z_j^i) \leq s_i$, and has an associated cardinality $|Z_j^i|$, i.e., it represents a parametric set of colors (in the sequel we shall consider cardinality one dynamic subclasses). It must hold, for each k : 1...s_i

$$\sum_{j:d(Z_j^i)=k} |Z_j^i| = |\, C_{i,k} \,| \qquad (1)$$

The token distribution in *sm* is defined by a function (denoted itself *sm*) mapping each place *p* to a multiset on the *symbolic color domain* of *p*, obtained replacing each C_i with D_i in $\mathcal{C}$(p).

Among several possible equivalent representations, the canonical representative provides SM with an univocal formal expression, based on a lexicographic ordering of dynamic subclass distribution over the net places.

4. A STATE-TRANSITION SEMANTICS FOR REFLECTIVE PN

On the light of what said in section 2, the behavior of a reflective PN model between any meta-level activation and the consequent shift-down is naturally described in terms of (stochastic) Petri net state-transitions.

A state m_i is simply an ordinary marking of the base-meta PN, the Petri net obtained by suitably composing the base-level PN (a GSPN) and the meta-level PN (a SWN). Then, letting *t* (*t* ≠ *shift-down*) be any transition enabled in m_i, according to the GSPN (SWN) firing rules, and m_j be the marking reached upon its firing, we have the labeled state-transition:

$$m_i \xrightarrow{\lambda(t)} m_j \qquad (2)$$

where λ(*t*) denotes the weight, or the exponential rate, associated with *t*, depending on whether *t* is timed or immediate.

There is nothing else to do but consider the case where m_s is a vanishing marking enabling the higher-priority pseudo-transition *shift-down*: then

$$m_s \xrightarrow{w=1} m'_o \qquad (3)$$

m'_0 being the marking of the new base-meta PN, obtained first by replacing the (current) base-level PN with the GSPN isomorphic to the reification marking (once it has been suitably connected to the meta-level PN), then by firing *shift-down* as it were a normal immediate transition.

Using the same technique for eliminating vanishing states as in the reduced reachability graph algorithm for GSPN (SWN), it is possible to build a CTMC from the labeled state-transition graph of the reflective PN model.

4.1. Recognizing equivalent base-level evolutions

The state-transition graph semantics just introduced precisely defines the (timed) behavior of a reflective Petri net model, but suffers from two evident drawbacks. First, it is highly inefficient: the state description is exceedingly redundant, comprising a large part concerning the evolutionary strategy, which is unnecessary to describe the evolving system.

The second concern is even more critical, and indirectly affects efficiency: there is no way of recognizing whether the system, during its dynamics/evolution, reaches equivalent configurations. Deciding about system's state-transition graph finiteness and ergodicity are major performance analysis issues that are strictly related to the ability of recognizing equivalent behaviors/evolutions of the modeled system. More generally, a number of techniques based on state-space inspection rely on this ability.

For example, it may happen that (apparently) different strategies cause in truth equivalent structural changes to the base-level Petri net (the evolving system), that cannot be identified by the definition of state provided before. The combined effect of different sequences of evolutionary strategies might produce the same effects. Even more likely, the internal dynamics of the evolving system might lead to reach equivalent configurations.

The above tricky question, that falls into a graph isomorphism sub-problem, as well as the global efficiency of the approach, are tackled by resorting to the peculiar characteristic of SWN: the symbolic marking notion. The color domains of the meta-level PN are built of color class *Node*, representing the base-level PN nodes (places plus transitions) at the meta-level. As concerns the reification-set, they are:

$$C\,(reif_N),\ C\,(reif_M),\ C\,(reif_\Pi) : Node$$
$$C\,(reif_A) : Node \times Node \qquad (4)$$

Class *Node* is logically partitioned into places and transitions. More precisely, *Node* is defined as:

$$\underbrace{\overbrace{p_1 \cup \ldots p_k}^{named_P} \bigcup Unnamed_P}_{\text{places}} \bigcup \underbrace{\overbrace{t_1 \cup \ldots t_m}^{named_T} \bigcup Unnamed_T}_{\text{transitions}} \qquad (5)$$

.

Symbols $\{p_i\}$, $\{t_j\}$ denote singleton static subclasses. Conversely, subclasses $Unnamed_P$ and $Unnamed_T$ should be normally large enough to be considered as logically unbounded repositories of anonymous places/transitions.

Behind there is a simple intuition: while some ("named") nodes, for the particular role they play, preserve their identity during base-level's evolution, and may be explicitly referred to during base-level's manipulation, others ("unnamed") are undistinguishable from one another. In other words any pair of "unnamed" places (transitions) might be freely exchanged on the base-level PN, without altering the model's semantics.

There are two extreme cases: $named_P$ ($named_T$) = Ø, and, on the opposite, $Unnamed_P$ ($Unnamed_T$) = Ø. The former meaning that all places/transitions can be permuted, the latter instead that all nodes are distinct.

The technique we use to recognize equivalent base-level evolutions relies on the base-level reification and the adoption of a symbolic state representation for the base-meta PN that, we recall, results from composing in transparent way the base-level PN and the meta-level PN.

First we have to set as *initial state* of the reflective PN model a symbolic marking (sm_0) of the base-meta PN instead of an ordinary one: any dynamic subclass of $Unnamed_P$ ($Unnamed_T$) will represent an arbitrary "unnamed" place (transition) of the base-level PN.

Because of the simultaneous update mechanism of the reification (section 2), and the consequent one-to-one correspondence between the current base-level PN and the reification at the meta-level (Capra and Cazzola 2007), we can state the following

Definition 1 (equivalence relation). Let sm_i, sm_j *be two states of the reflective Petri net model.*
$sm_i \equiv sm_j$ *if and only if their projections on the reification-set have the same canonical representative.*

Consider the very simple example in figure 2, that depicts three base-level PN configurations, at different time instants. The convention we adopt is that while symbols t_2 denotes a "named" transition, symbols x_i and y_j denote "unnamed" places and transitions, respectively. In other words (abusing notation) x_i and y_j will denote also dynamic subclasses of $Unnamed_P$ and $Unnamed_T$, respectively. We assume that all transitions have the same priority level.

We can observe that the base-level PNs on the top and on the middle have the same structure, but (apparently) different current marking. We can imagine that they represent a possible (internal) dynamics of the base-level Petri net.

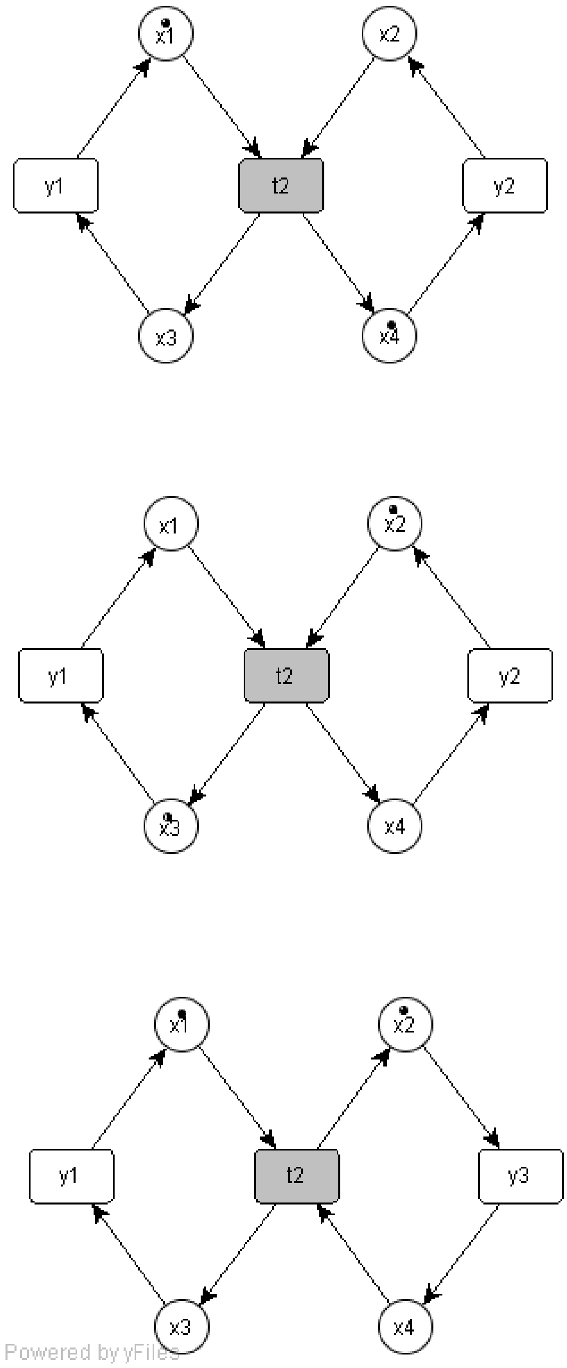


Figure 2: Three equivalent base-level PN

Conversely, we might think of the Petri net on the bottom of figure 2 as an (apparent) evolution of the base-level PN on the top, in which transition y_2 has been replaced by a new transition (y_3), new connections are set, and a new marking is defined.

Nevertheless, the three base-level configurations are equivalent, according to definition 1. It is sufficient to take a look at their respective reifications, that are encoded as symbolic markings (hereafter multisets are expressed as formal sums).

Consider first the base-level PNs on the top and on the middle of figure 2, whose reification are:

$$\begin{aligned}
&sm(reif_N) = y_1 + y_2 + t_2 + x_1 + x_2 + x_3 + x_4\\
&sm(reif_M) = x_1 + x_4\\
&sm(reif_A) = <x_1, t_2> + <t_2, x_3> + <x_3, y_1> + <y_1, x_1> +\\
&<x_2, t_2> + <t_2, x_4> + <x_4, y_2> + <y_2, x_2>
\end{aligned} \tag{6}$$

and

$$\begin{aligned}
&sm'(reif_N) = y_1 + y_2 + t_2 + x_1 + x_2 + x_3 + x_4\\
&sm'(reif_M) = x_3 + x_2\\
&sm'(reif_A) = <x_1, t_2> + <t_2, x_3> + <x_3, y_1> + <y_1, x_1> +\\
&<x_2, t_2> + <t_2, x_4> + <x_4, y_2> + <y_2, x_2>
\end{aligned} \tag{7}$$

respectively.

They can be obtained from one another by the following permutation of "unnamed" places and transitions (we denote by $a \leftrightarrow b$ the bidirectional mapping: $a \rightarrow b$, $b \rightarrow a$):

$$\{x_1 \leftrightarrow x_2, x_3 \leftrightarrow x_4, y_1 \leftrightarrow y_2\} \tag{8}$$

hence, they are equivalent.

With similar arguments we can show that the base-level PN on the top and on the bottom of figure 2 are equivalent too. The bottom's Petri net reification is:

$$\begin{aligned}
&sm''(reif_N) = y_1 + y_3 + t_2 + x_1 + x_2 + x_3 + x_4\\
&sm''(reif_M) = x_1 + x_2\\
&sm''(reif_A) = <x_1, t_2> + <t_2, x_3> + <x_3, y_1> + <y_1, x_1> +\\
&<x_2, y_3> + <y_3, x_4> + <x_4, t_2> + <t_2, x_2>
\end{aligned} \tag{9}$$

sm and *sm''* can be in turn obtained from one another by the following permutation:

$$\{x_2 \leftrightarrow x_4, y_3 \leftrightarrow y_2\} \tag{10}$$

The canonical representative for these three equivalent base-level PN's reifications (i.e., states of the reflective PN-model), computed according to the corresponding SWN algorithm, turns out to be *sm*.

5. CONCLUSIONS AND FUTURE WORK

We have semi-formally presented a (timed) state-transition graph semantics for reflective Petri nets, a formalisms well suited to model evolvable discrete-event systems, based on classical stochastic Petri nets (GSPN, and their high-level version, SWN). In particular, we have addressed major topics related to recognizing equivalent system's evolutions, by exploiting the SWN's symbolic state notion. We are planning to integrate the GreatSPN tool, that natively supports GSPN and SWN, with new modules for the graphical editing of reflective PN models, and their analysis/simulation based on the associated state-transition semantics. We are also investigating possible applications of reflective Petri nets for the analysis of dynamic workflows (already specified using the same formalism) and protocols for mobile ad-hoc networks.

ACKNOWLEDGMENTS

The author thanks Walter Cazzola, who contributed to the definition of Reflective PN, for its valuable suggestions .

REFERENCES

Capra, L., 2008. Addressing soundness and efficiency issues in dynamic processes: a reflective pn-based modeling approach. *In Proceedings of SCS Spring Simulation Multiconference (SpringSim'08) - Business and Industry Symposium*, April 14 – 17, Ottawa (Canada).

Capra, L., and Cazzola, W., 2007. Self-evolving petri nets. *Journal of Universal Computer Science*, 13(13): pp 2002-2034, available from: http://www.jucs.org/jucs_13_13/self_evolving_petri_nets [accessed 17 Dec 2007]

Chiola, G., Dutheillet, C., Franceschinis, G., and Haddad, S., 1993. Stochastic Well-Formed Coloured Nets for Symmetric Modelling Applications. *IEEE Transactions on Computers*, 42(11):pp 1343–1360.

Chiola, G., Franceschinis, G., Gaeta, R., and Ribaudo, M., 1995. GreatSPN 1.7: Graphical editor and analyzer for timed and stochastic Petri nets. *Performance Evaluation*.

Marsan, M. A., Balbo, G., and Conte, G., 1983. A class of generalised stochastic petri nets for the performance evaluation of multiprocessor systems. *In SIGMETRICS'83: Proceedings of the ACM SIGMETRICS conference on Measurement and modeling of computer systems*, pp 198–199, New York (NY, USA).

AUTHORS BIOGRAPHY

Lorenzo Capra was born in Monza (Italy), and went to the University of Milan, where he obtained his Laurea degree in Computer Science in 1992. After having collaborated for several years with the Automation Research Center at the National Electric Power Provider (ENEL), he moved to the University of Turin, where he received a Ph.D in Computer Science. He is currently assistant professor at the Dept. of Informatics and Communication (Di.C.O) at the University of Milan. His research interests include High-Level Petri Nets analysis/simulation and formal methods in software engineering.

ANALYSIS, MODELING AND SIMULATION OF THE INCORPORATION OF EGOVERNMENT IN ADMINISTRATIVE PROCESSES

Pedro Baquero[a], Yurena García-Hevia [b], Rosa María Aguilar [c]

[a] Department of F.F.&E, Electronics and Systems, University of La Laguna, Spain
[b] Engineer at the Computer Science School, La Laguna University, Spain
[c]Department of Systems Engineering and Automation and Computer Architecture, La Laguna University, Spain

[a] pbaquero@ull.es, [b] ygarcia@isaatc.ull.es, [c]rosi@isaatc.ull.es

ABSTRACT

In this paper we present a general model for analyzing the incorporation of eGovernment in the administrative processes that take place in a public institution. This model is developed in three conceptual visions which allow for advances in administration modeling. We start with an organizational vision and proceed to describe the operational vision (i.e., of the processes), and conclude with a technological vision that conceptualizes the processes in an architecture that is able to be simulated. Lastly, we detail a simple test case which analyzes the results of incorporating electronic processes into the services offered by an administrative agency.

Keywords: egovernment; administrative processes, simulation

1. INTRODUCTION

In recent years, public agencies have experienced significant advances from the impetus of the Information Society. To this end, in Spain, Law 11/2007, dated 22 June, on the electronic access to Public Services by citizens, was enacted for the purpose of laying the policy and legal foundations for allowing the citizen to file his paperwork with various agencies either in person or electronically. Among the requirements of this law is that the application of electronic means to handle procedures, processes and services be preceded by the performance of a functional redesign analysis and a simplification of the procedure, process or service. The following aspects are of particular significance: a) the elimination or reduction of the documentation required of citizens, and its substitution with data, data transmissions or certificates, or the regulation of the processing once the transaction is complete; b) an allowance for means and instruments of participation, transparency and information; c) a reduction in response times; and d), the rationalization of the work load distribution and of internal communications.

Within this context, the Government of the Canary Islands has promoted a project for the Interoperability of its Electronic Services intended to modernize its electronic infrastructure. This entailed an analysis of the organization to come up with a valid eGovernment model that would help achieve the primary objective, which was to gradually obtain an efficient model which would allow the Government to adapt to the number of conceptual, organizational and technological changes that still lie ahead. So those aspects intended to streamline and rationalize the process necessarily imply a reduction in the time required to resolve administrative procedures, which in turn result in reduced effort on the part of both the citizen and the agency. This requires that, as part of the planning process for the implementation of electronic procedures, a forecast be made of the expected improvements not only in the filing process for the citizen, but in the productivity of the agency.

This paper presents the framework used for the modeling of administrative process so as to yield a technological model that can be effected to simulate these processes. Section 2 presents the three conceptual visions of the model implemented. The starting point for the modeling process is a hierarchical scheme that involves three models: organizational, operational and technological. These models are described in Sections 3, 4 and 5 respectively. The organizational model merely considers the functional requirements of the simulation, and contains the description of the organization that is the object of the analysis. The operational model addresses the processes. In this paper we will focus only on those processes involving administrative procedures and will not consider the analysis of other types of non-structured processes. The technological model focuses on constructing an architecture that can be implemented in the simulation. Section 6 describes the simulation run on a test case involving how the Government of the Canary Islands oversees the Gaming Board. Lastly, the conclusions drawn from our research are presented.

2. HIERARCHICAL MODEL OF THE ADMINISTRATIVE PROCESSES

A public agency is an organization consisting of a multitude of units executing processes. We refer to the minimum organizational unit that performs a specific process as an operational unit. Said unit comprises a set of resources. The processes are executed by human resources and/or tools (e.g. computer applications) based on set procedures (generally written).

The structural relationship between operational units defines this organization's process architecture. If we consider the modeling of processes as related to administrative procedures, then the model encompasses three conceptual visions:

1. The organizational vision represents (a) the overall view of the relationships among all the operational units, (b) the strategic vision and the services the operational unit offers its clients, (c) a description of the organizational structure and of its associated resources, functions and procedures, and (d) the standards for evaluating and monitoring the operational results.
2. The operational vision defines how the entire organization's processes are carried out as one, as well as each operational unit's specific processes. The operational model describes: (a) the operations which support the organization's strategic mission; (b) the process necessary to provide each service in an operational unit; (c) the management of resources, both internal and external, used to provide the services.
3. The technological model defines how specific aspects of the organizational and operational model should be handled in the final implementation of the model and simulation of the ICT infrastructure. In the technological model we define mainly the general architecture of the entire organization, as well as the specific architecture of each operational unit.

The organizational and operational models form the theoretical reference model of the ICT infrastructure. The former considers the infrastructure as a static organizational model, and the latter as a dynamic organizational model. The technological model, on the other hand, translates these theoretical model into logical patterns that can be implemented using computational tools. A brief description of each of these models is given next.

3. ORGANIZATIONAL MODEL

At the first level of the organizational vision, the operational units are seen as black boxes such that certain inputs produce certain outputs.

The organizational model contains the description of the set of procedures grouped by services, which in turn are grouped by areas. In general, there are two decision grouping levels, the more specific one being found in the service grouping associated with executive level decisions. The less specific one is found in the area or theme grouping, and corresponds to policy decision levels. In the former, the decisions aim for administrative efficiency, while in the latter they are adapted to the policies of each governing body.

The highest level of precision contains the procedure definitions. These arise from an analysis of the legal guidelines regulating the services to be provided to citizens. In general, all of these procedures are set, there being implicitly or explicitly just one way to resolve a citizen's request for service.

The resource level is also developed within this model as conditioned by the agency's organizational structure, along with the functions they carry out. This model also considers the set of policies and projects necessary to implement the different computer applications which gradually allow for the incorporation of eGovernment. In this model we propose new ways to resolve service requests by using computational tools.

The organizational model also describes the target public, divided by services.

Lastly, the objectives to be met by each of the services are defined in the organizational model. For the case in question, these objectives are oriented at improving interactions with the customer by reducing the response and turn-around times and the rationalization of the work load distribution.

4. OPERATIONAL MODEL

The operational model defines the way the services are be apportioned to the resources being managed within the operational units. It also controls the mutual interaction between the operational units. The operational model translates the description of the elements and procedures of the organizational vision; that is, it introduces the element of time into the static representation of the organizational model. This model is thus better suited for modeling organizational processes.

The policy level is defined at the first level and uses a system for prioritizing certain policies over others, clarifying the prioritization of some services over others. The processes taking place inside the operational units are described at the second level. The process result from applying the procedures defined in the organizational model.

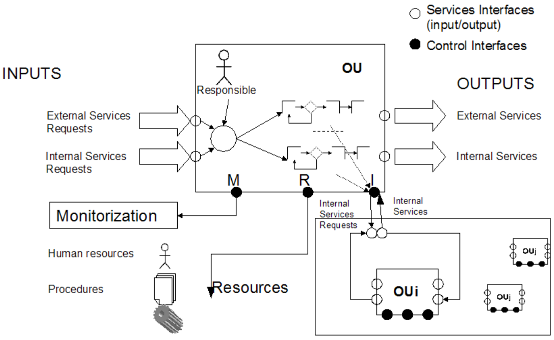


Figure 1: Operational model of an operational unit

Figure 1 shows a general representation of the process model for an operational unit. Basically, it shows how an operational unit (OU) accepts service requests (SR) from other OUs and from the environment (users). These SRs are prioritized internally for subsequent execution of the requests by the resources available to the OU. The resources are of two types, the first ones are human and tools (which execute the processes) and the second one are procedures (which specify how to execute the processes). The processes can, in turn, call other OUs to request services. In other words, an OU can carry out processes obtained from other OUs and for which they are not directly responsible. This means that in certain cases, conflicts may arise when deciding which process to prioritize. The policy level is charged with communicating to each OU which services take priority.

Each OU's resources and services are obtained from the organizational model. In addition, there is an interface which monitors the processes in accordance with the standards defined in the organizational model.

5. TECHNOLOGICAL MODEL

This model features an architecture intended for modeling and simulating processes. There are two architectural levels: one in which the organization's general architecture is represented, and another where the operational unit's specific architecture is represented.

5.1. General architecture

Figure 2 shows the technological model of the organization. Note the presence of an orchestrator (which represents the policy decision levels), responsible for executing the services depending on the organization's priorities, that is, prioritizing some services over others.

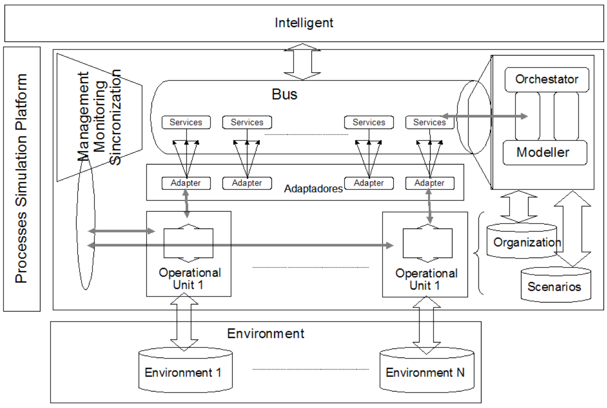


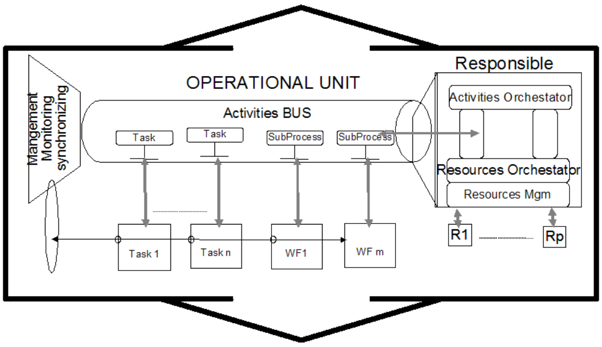
Figure 2: Overall architecture of an organization

The Orchestrator is charged with defining the different scenarios that can exist (for example, intensive vs. scarce use of the ICT by the users). The Orchestrator can also gauge each of the organization's available resources (for example, the introduction of electronic applications, assignment of human resources to the different OUs). The bus, in addition to providing communications between the OUs, is charged with publishing the different services available. A certain algorithm is used to assign the priority each service has to the organization. The modeled setting is also represented in a database, which is the set of citizens that can make use of the agency's services. Lastly, there is a unit charged with managing, monitoring and synchronizing the platform and whose purpose is the proper operation of the simulation platform.

5.2. OU Architecture

Figure 3 shows the technological model of the OU. We can see that there is another orchestrator, representing the manager of the executive level decisions. This orchestrator ensures that the processes are executed according to the priorities established by the policy levels, that is, it prioritizes certain processes over others.

Figure 3: Specific architecture of an operational unit



In this case, the Orchestrator allots, either manually or automatically, the resources to the activities to be executed. It also features its own internal control scheme so as to have a specific platform available for each OU.

The simulator uses different scenarios defined by the policy level.

6. SIMULATION (TEST CASE)

For this paper, we implemented and simulated the services associated with an office that offers services specific to the Gaming Board in the Canary Islands. The Gaming Board manages procedures associated to casinos, games of chance and arcade games.

6.1. Test case

In this subsection the organizational model is described. In this case, there are two OUs: one associated with the Game Management (GM), and the other with the Game Inspection (GI). An additional OU is used for receiving the service requests, the Input Register for the Government of the Canary Islands (IR).

The GM mainly receives service requests from the citizens. These requests are received by the IR. In addition, the GM needs certain services provided by the GI, such as a verification of specific actions to be carried out by the citizens. The GI must also perform inspections, whether they be occupational or resulting from citizen complaints.

The GM prioritizes the service requests according to technical requirements, such as (legally binding) response times. The GM also prioritizes its inspection processes depending on policy requirements, such as prioritizing those processes associated with GM service requests above those arising from customer complaints, or vice versa.

The set of procedures associated with the GM and GI is fairly broad, the former having 52 and the latter 10 procedures. Each of these procedures has a specific volume of service requests. Some services are in constant demand, while others are seasonal. For example, a procedure called "removal due to substitution" may have 2000 requests spread evenly throughout the year, while the procedure called "exchange of slot machine" may have an isolated demand of 60 requests concentrated in the month of December.

Each procedure has an in-person and an on-line request associated with it. The first assumes the citizen does not make use of the electronic method to present his service requests. Different demand percentages for each of these types of processes are set in each scenario.

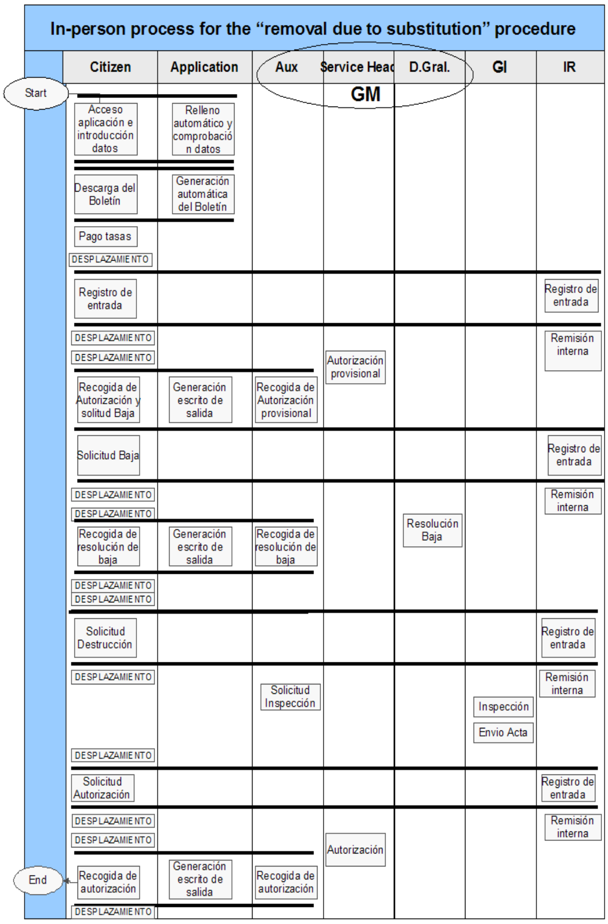

In-person process for the "removal due to substitution" procedure
Citizen
Application
Aux
Service Head
D.Gral.
GI
IR
GM
Start
End
Acceso aplicación e introducción datos
Relleno automático y comprobación datos
Descarga del Boletín
Generación automática del Boletín
Pago tasas
DESPLAZAMIENTO
Registro de entrada
Remisión interna
Autorización provisional
Recogida de Autorización y solitud Baja
Generación escrito de salida
Recogida de Autorización provisional
Solicitud Baja
Recogida de resolución de baja
Recogida de resolución de baja
Resolución Baja
Solicitud Destrucción
Solicitud Inspección
Inspección
Envio Acta
Solicitud Autorización
Autorización
Recogida de autorización

Figure 4: In-person process for the "removal due to substitution" procedure

6.2. Objectives

The objective of the use of simulation is to study, based on different scenarios:

- The average time to resolve the different procedures as the citizen makes use of the electronic method.
- The average decrease in the citizen's effort when the electronic method is implemented.
- The decrease in the human resources of the Government of the Canary Islands when the electronic method is implemented.

6.3. Simulation tool

The above considerations were addressed in this paper so as to yield a modeling and simulation environment for the services of a public agency. In this context, the simulation is presented as a tool to help management with the decision-making process (Hansem 2006). Through the simulation, the question of "what if" can be answered for different scenarios by checking the effectiveness of the possible actions to be applied.

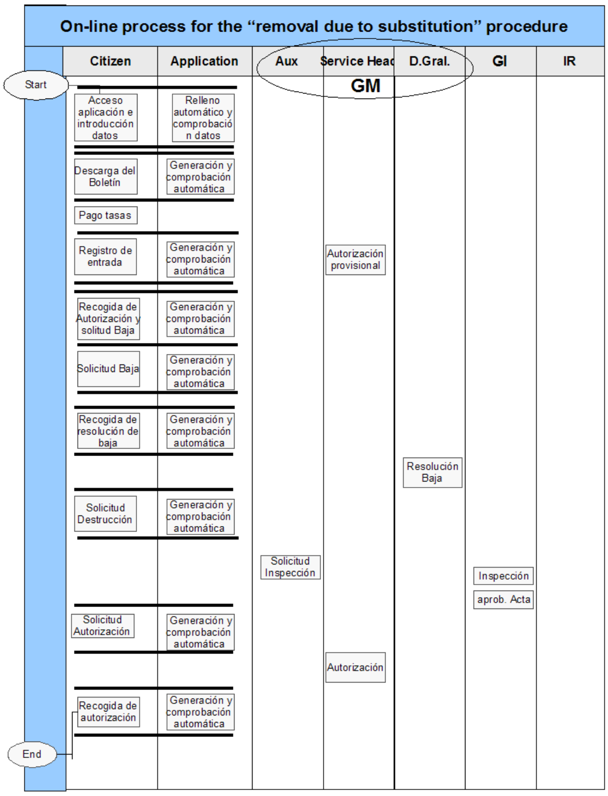


Figure 5: On-line process for the "removal due to substitution" procedure

The tool used was the Java simulation library for discrete event systems named SIGHOS (SIGHOS 2007), and specifically its XML interface (XMLGHOS) which facilitates the use of the library by introducing a layer that abstracts low-level simulation mechanisms (Aguilar et al 2006). Both tools are provided as freeware (http://sourceforge.net/projects/sighos/), and were developed by the Simulation Group at the Systems Engineering and Computer Automation, Architecture and Technology Department of the University of La Laguna.

The library uses a process-oriented methodology to model a system (Lorenzo 2001). In other words, the system is characterized by having the elements flow in stages which can vary depending on the status of the element itself. In one stage, these elements carry out different activities, and in doing so utilize system resources, meaning they have to wait for said resources to become available, at which time they will engage the resource for as long as it takes to perform the activity. This is why it is vital to specify when a resource is available and what functions it can perform.

6.4. Implementation

The first step consists of specifying the processes in accordance with the diagram in the figure. This is done by defining all the processes, along with the priorities and requirements to be met concerning time and importance. Figures 4 and 5 show an example for the same procedure for the two processes associated with in-person and on-line events.

Thanks to the modular design of the SIGHOS library, each module can be mapped to the architecture in Figures 2 and 3. The simulations are described simply by using XGHML (which is the XML SIGHOS Modeling Language).

The system to be simulated is described using two files, one referred to as MODEL, in which the resources, activities and flows followed by the elements are described, and the other as EXPERIMENT, in which the number of simulations desired of a model, the type of information to be displayed and the types of elements flowing in the model are specified.

For example, the simulation of the flow of service requests will allow us to determine how response times evolve. To do so we have to convert the flow diagram to an algorithmic model. The different resource types (resourceType), its characteristics (resource), the activities (activity), the meta-flows (rootFlow) in the system and the types of elements (elementType) that will flow through it are described in the model using XGHML.

The types of resources are level 1 agents (which directly handle the service requests) and level 2 agents (which authorize a petition or provide value services). The IR unit only has level 1 resources. The GM has level 1 and 2 resources at its disposal, while the GI only has level 2 resources.

Defining the resource types is rather simple. The resource need only be assigned a unique key and a brief description.

```
<!-- Resource types -->
<resourceType id="1">
  <description>Level 1 Agent</description>
</resourceType>
```

The resource type descriptions are defined with the resource label, which is used to specify the units to be generated, when each is available, how often this resource is generated and how long the turn lasts. In the following example, a level 1 agent is generated which works every day from 8 am until 4 pm.

```
  <!-- Resource description -->
  <!-- LEVEL 1 AGENTS -->
  <!-- Day shift (8:00 - 16:00) -->
  <resource id="1">
    <description>Level 1 Agent</description>
    <units>1</units>
    <timeTable>
      <rt_ref model_id="1" id="1"/>
      <cycle>
                      <timeUnit>HOUR</timeUnit>
<!-- Time shift begins -->
        <startTs>8.0</startTs>
        <iterations>0</iterations>
        <period>
          <dist>FIXED</dist>
<!-- Each day -->
          <p1>24.0</p1>
          <p2>0.0</p2>
          <p3>0.0</p3>
        </period>
      </cycle>
                <dur  timeUnit="HOUR">8</dur>
<!-- 8-hour shifts -->
    </timeTable>
  </resource>
```

Currently there is no module for representing the orchestrators for both the global architecture and the specific operational unit architecture. The priorities in this case have to be manually assigned to the activity descriptions. The activity label is used to describe the activities, which must have a unique identifier assigned. The label accepts a brief description of the activity and allows priorities to be assigned if several activities will be executed at once (not used in this project). A working group is then assigned in which the resources and the time necessary to complete the activities are specified.

In this case, an “Input Register” is described, and involves an in-person activity which must be executed by a level 1 agent and has a duration of 21 seconds.

```
  <activity id="1">
<description>BasicData
Registration</description>
    <priority>0</priority>
    <presential>true</presential>
    <workGroup id="1">
      <description/>
      <role>
        <rt_ref model_id="1" id="1"/>
        <units>1</units>
      </role>
      <priority>0</priority>
      <duration>
        <dist>FIXED</dist>
        <p1>21</p1>
        <p2>0.0</p2>
        <p3>0.0</p3>
      </duration>
    </workGroup>
  </activity>
```

The types of elements in this model are the different service requests that may appear depending on the associated procedures.

The element flows through the activities are described in the meta-flows, which are implemented using the tools provided by XGHML. A definition based on meta-flows allows the different operational units to be modeled.

Lastly, a group of listeners is implemented. These listeners indicate what measures are used for the resolution times for each service request, and the utilization times for each citizen and resource.

6.5. Results

The results of the simulation are tabulated in Table 1. Columns show scenarios with different relations between procedures resolved in-person and on-line. These quantitative measures are consistent with the qualitative results expected, as evidenced by the significant reduction in the resolution times in all the scenarios, along with a decreased effort on the part of the citizen, this decrease being particularly noticeable in the third scenario.

The advantage of this modeling environment is that it allows for a quantitative prediction involving different scenarios when a large quantity of procedures are implemented.

Table 1: Results of simulations

	In-person	Online	In-person	Online	In-person	Online
Scenarios	100,00%	0,00%	50,00%	50,00%	20,00%	80,00%
Average Time Procedure (days)	5,2	100,00%	2,63	50,64%	1,09	21,03%
Average Effort/procedure (min):						
Citizen	166,5	100,00%	96,25	57,81%	54,1	32,49%
Level 1 GM	8,7	100,00%	4,9	56,32%	2,62	30,11%
Level 2 GM	3,4	100,00%	3,05	89,71%	2,84	83,53%
Level 2 GI	13,2	100,00%	13,15	99,62%	13,12	99,39%
Level 1 IR	9,2	100,00%	5,1	55,43%	2,64	28,70%

7. CONCLUSIONS

This paper described the implementation of a modeling and simulation environment that allows for quantitative predictions of process improvements when an electronic method is introduced for a specific service. To test this enviroment only 50 procedures for a given service involving three operational units were implemented in this simulation (in the Government of Canary Islands there are about 5000 procedures). As all the procedures for all the services are introduced, a much more global measurement of the total implementation policy for eGovernment will become available.

The modeling environment used attempted to represent the architectures described in this paper. This was done by using the different modules that comprise the SIGHOS library and simulating specific functionalities, such as that of the orchestrator. To date we have not implemented a fixed environment that allows for a global configuration. Work is ongoing to introduce artificial intelligence techniques in the development of the orchestrators.

The implementation of all the procedures of a public agency, such as that associated with the Government of the Canary Islands, poses significant challenges. These challenges result from the large quantity of processes, possibly in excess of 5000, to be modeled. Since modeling the process associated with one procedure can require in the best of cases one full working day, modeling all the processes would be an imposing task indeed, hence the efforts toward making available an environment which can translate the application model directly into a process model. To this end, we are waiting for the Government of the Canary Islands to define or select a standard language for modeling its procedures so that an application code to process code converter can be developed.

ACKNOWLEDGMENTS

This work is being supported by a project (reference DPI2006-01803) from the Ministry of Science and Technology with FEDER funds.

REFERENCES

Aguilar, R. M., Muñoz, R., Castilla, I., Martín, C. A., Muñoz, V., 2006. An interface to integrate workflow diagrams into discrete event simulation. *International Mediterranean Modeling Multiconference*. Barcelona-España..

Hansen, M. H.; Netjes, M., 2006. Business process simulation - a tool survey. *Seventh Workshop and Tutorial on the Practical Use of Coloured Petri Nets and the CPN Tools,* 579, 77-96.

Moreno, L., Aguilar, R.M., Piñeiro, J.D., Estévez, J.I., Sigut, J.F., González, C., 2001. Using KADS methodology in a simulation assisted knowledge based system: application to hospital management, *Expert System with Applications*, vol. 20, 235-249.

SIGHOS Home Page, 2007. Available from http://sourceforge.net/sighos

AUTHORS BIOGRAPHY

PEDRO JUAN BAQUERO PÉREZ, is a Telecommunication Engineer from the Polytechnic University of Madrid. He is an associate professor at University of La Laguna where his research is focused on the modelling and simulation of IT organizations Also, currently he is Head of the Unit for Information Technology of the Ministry of Presidency and Justice of the Canary Island Regional Government. He has worked in different areas: as a freelance consultant in information technology, as an researcher (Interuniversitary Microelectronic Center (Belgium) and Telefónica I+D), as a university professor (University of Carlos III of Madrid), as a civil servant (telecommunication specialist in the Spanish State Telecommunications Office and Head of the Telecommunications Unit in the Government of Canary Islands), and as a managing director (Head of Strategic Planning Area in a cable telecommunications operator and Head of Costumer Engineering Area in a IT company).

YURENA GARCÍA-HEVIA MENDIZÁBAL is a Technical Computing Systems Engineer from the University of La Laguna. She is currently a student of Engineering Computer Science and a research grant holder at the Department of Systems Engineering and Automation at the University of La Laguna. Her fields of interest are the design and implementation of simulation environments for service organizations, modeling and simulation, and business intelligence.

ROSA M. AGUILAR received her MS degree in Computer Science in 1993 from the University of Las Palmas de Gran Canaria and her PhD degree in Computer Science in 1998 from the University of La Laguna. She is an associate professor in the Department of Systems Engineering and Automation at the University of La Laguna, Canary Islands, Spain. Her current research interests are decision making based on discrete event simulation systems and knowledge-based systems, intelligent agents, and intelligent tutorial systems.

SIMULATION OF TRUST IN CLIENT – WEALTH MANAGEMENT ADVISOR RELATIONSHIPS

Terry Bossomaier[(a)], Russell K. Standish[(b)]
[(a)(b)]CRiCS Centre for Research in Complex Systems, Charles Sturt University.
[(b)]School of Mathematics and Statistics, University of New South Wales.
[(a)]tbossomaier@csu.edu.au, [(b)]hpcoder@hpcoders.com.au

ABSTRACT

This paper describes a two phase model for simulating trust amongst clients and their wealth management advisors. In phase one an artificial life model was used to assess the dynamics of trust. In phase two the model is extended to utlise real data from a corporate database of client information. The alife model highlighted needs for information not captured directly, requiring sophisticated inference techniques. Fuzzy logic is used to describe client behaviour with rules found through evolutionary optimisation. Analysis of mutual information between time series of clients investments is used to determine links between clients.

1. INTRODUCTION

Today's companies operate in a very complicated and sometimes turbulent environment. Unexpected changes in global resources such as the food crises in 2008 and the escalating oil price can dramatically change market positions. On the other hand the subprime mortgage meltdown has revealed just how complicated and fragile financial systems can be.

Decision support systems and mining of the vast quantities of consumer data in corporate data warehouses are valuable but their capacity to predict future requirements is often limited to narrow extrapolation from the past. More powerful scenario planning systems are needed which can explore new trends, such as, for example, the rapid switch of agricultural land from food to biofuel production, a powerful influence with little prior history.

Agent based models (ABMs) attempt to model people, companies and external forces and to go beyond simple extrapolation. From early minimalist models, some ABMs now incorporate many millions of agents, such as the large Epicast models for studying pandemics in the USA.

Until a few years ago Australian workers were in general locked in to the superannuation fund provided by their employer. But following deregulation, people were allowed to choose to which of many available funds they belonged. Inevitably this created a significant market for financial advisors. At first such services were poorly monitored and advice was not always sound. Worse there were a number of concerns about conflict of interest, where advisors were given hidden trailing commissions or promoted a fund owned by their parent organisation.

Thus trust in financial services became an important issue for banks and other providers to confront. This project focussed on building an ABM with two distinct goals

1. to examine the dynamics of trust, to look for phase transitions and indicators of declining trust. This was essentially an abstract artificial life model.
2. to build a realistic agent based model derived from real data.

The artificial life model (Bossomaier, Jarratt, Anver, Thompson & Cooper 2005) consisted of sets of clients and wealth management advisors (WMAs) and an artificial stock market. The advisors compete with one another to maximise their wealth, while the clients share trust information with one another.

The stock market simulates shares or share portfolios, which are owned by the client and are paid for with upfront consultancy fees, and funds, aggregates of shares from the same market, which pay commissions. The growth of the shares follows a deterministic equation with added noise. Advisors may purchase access to the growth

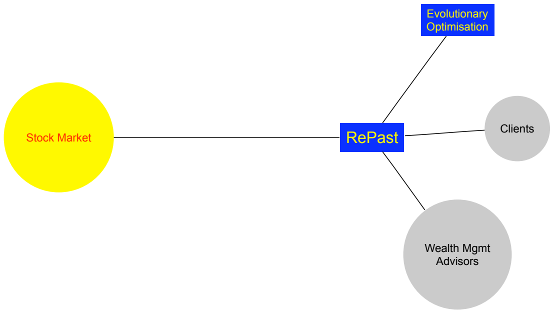


Figure 1: Architecture of the trust ALife model

parameters of this equation, thus trading off costs against better advice to clients.

Clients trust evolves according to their returns on investment, using insights from the neuroeconomics literature. Each client is connected to other clients on one of a number of different types of network, lattice, small world and scale-free. We present results for the wealth of clients and advisors for different networks and different research costs.

To move to the full scale model involves analysing a large corporate database of several million clients. The first stage in doing this requires analysis of the client network but this is not available from any fields in the database itself. Thus indirect methods are needed to infer the connectivity. A novel approach has been adopted which involves looking for common patterns in investment over time among clients.

2. THE ALIFE MODEL

The ALife model consists of several components as shown in figure 1.

Wealth Management Advisors invest money on behalf of their clients taking some profit along the way. They invest the clients' money in a mixture of shares and funds[1]. Shares return a percentage of the investment as a once off fee. Funds provide an ongoing trailing commission.

Clients split their wealth into a fraction invested with their WMA and leave the rest in the bank, the fraction being determined by their trust level. If their trust falls significantly below the trust of their neighbours,

The stock markert is not intended to represent the real stock market in any detail and the many diverse investment options it provides. It merely provides an investment framework.

Simulations of this model allowed the study of the evolution of trust under various conditions. One important issue is the nature of the client networks. If the clients are unconnected then their trust will go up and down with WMA performance and the WMA can trade off the additional investment he gets with increased trust against the potential loss of fees and commissions. But if the clients can talk to each other then the WMAs now have to outperform each other to avoid loss of clients to others who provide better net returns.

Much recent interest in networks has led to three commmon types in social systems: simple local connectivity, such as a lattice; small world networks in which additional long range connections are added (Watts 1999); and scale free networks characterised by a power law distribution in the connectivity of nodes (Barabási 2002). Different social networks will lead to different trust flow behaviours, hence modelling these networks is important. However, this information may not be readily available.

2.1. The Stock Market

There are very many stock market models aournd in the literature and this paper neither tries to improve on them, nor to evern select the best. Two factors drive the approach

- the model must be efficient in comuting resources
- it must have a natural transition to real financial instruments or products and must reflect investment in research activity into the value of different sorts of investment.

A simple model satisfying these requirements is used data from the NY Stock Exchange obtained via Yahoo finance (Yahoo 2008). An exponential fit to this data was lperturbed by an additional noise term using Brownian motion as in equatiion 1.

$$y = A_0 exp(a_1 * t + a_2 * c) \quad (1)$$

where A_0, a_1, a_2 are constants and c is a cummulative uniformly distirbuted random number in the range $[0 - 1]$.

---

[1] To make the simulation computationally tractable, portfolioss are relatively small and each share may be thought of as more an assest class than an individual equity.

The full bank model uses the financial instruments constructed by the bank and their variation over time as measured on the real stock market.

3. THE DATASETS

A large dataset of 456 million records describing 14 million customers over a five year period, and a smaller more detailed dataset of 42 million records describing the investment profiles of 1.5 million customers over a three year period were supplied to the project.

The larger dataset's records contained demographic details, as well as aggregate account balances over the three categories of cash, loans and investments. Since these account balances will depend in an intricate way upon the account product performance, customer income and expenditure, details which are not available in the dataset, it was decided to model the discrete events when existing customers enter or leave a particular account class. This includes (for example) existing cash or investment customers taking out a mortgage, or investors closing all their investments accounts, which is of particular interest to the trust project.

Customer behaviour is represented by values drawn from $\{-1, 0, 1\}$ where -1 indicates that all accounts of the class are closed, 0 means no change and 1 indicates that that customer has entered that class. Thus three timeseries are available in the dataset, representing the behaviour in the three classes. A fourth timeseries is generated from change in the number of dependents, perhaps due to the birth of child, or through marriage, or conversely through children growing up and leaving home.

From the smaller dataset, it was possible to establish the investment profiles of the 1.5 million investment customers. Whilst product performance was not available in this database, it was possible to match the internal product identifiers to published product information, and to download the relevant product performance from the bank's website. By defining a product's risk as the variance of its performance (historical volatility), one can estimate a customer's risk profile by taking the product balance weighted average of the product's risk. This will fluctuate over time as the product balances change (unless the customer is invested in a single product only), but if the customer performs active portfolio balancing, this will reasonably accurately reflect the customer's risk preference. The resulting timeseries has six independent variables (age, length of customer relationship, gender, marital status, deceased and number of dependents), and one dependent variable (risk).

4. MODELLING CLIENT BEHAVIOUR

Human behaviour may be represented in several ways, each extremely diverse (Fulcher 2008). Artificial neural networks (ANNs) are loosely linked to the structure and operation of the human brain but there are very many architectures and training or learning algorithms from which to choose. At the other extreme to ANNs are formal rule based systems. But representing human behaviour with rules is tricky, often requiring very large rule sets. Yet these two extremes ultimately have to converge to the same outcomes since both are capable of arbitrarily accurate representations.

Fuzzy logic falls somewhere in between. Its advantage in ABMs arises from the interdisciplinary nature of socio-economic modelling. Qualitative research outcomes and judgements from domain experts can be readily transcribed into fuzzy logic and its conception was in part motivated by the semi-quantitative style of much human thinking. Thus fuzzy logic is the methodology used herein.

4.1. Fuzzy Inference Systems

Fuzzy sets are sets whose elements have a degree of membership in the range $[0, 1]$. More precisely, a fuzzy set F is a pair $F = (A, m)$, where A is a set, and $m : A \to [0, 1]$ is the membership function. If $m(x) = 0$, then x not considered to be a member of the fuzzy set F, and if $m(x) = 1$ then x is considered fully included in the fuzzy set.

Fuzzy logic extends the notion of propositional logic to fuzzy sets with the fuzzy logic operators AND, OR and IS. Fuzzy logic rules are of the form:

$$\begin{aligned} &IF\ x_1\ IS\ I_{11} \\ &\quad AND\ (x_2\ IS\ I_{21}\ OR\ x_2\ IS\ I_{22}) \ldots \\ &\qquad THEN\ y\ IS\ O_1. \end{aligned} \tag{2}$$

There are a variety of *fuzzy inference systems* (FIS). The *Mamdani* type, used here , consists of a number of input variables x_i, whose ranges are partitioned into fuzzy sets I_{ij}, an output variable y whose range is partitioned into fuzzy sets O_j, and a set of fuzzy rules of the form (2). The FIS takes a vector of input values, and outputs an inferred value $\hat{y}$.

The *matching degree* $w(x_1, x_2, ...x_n)$, or *weight*, for a rule is constructed from the antecedent, where the IS operator is replaced by the membership function, AND is replaced by a binary operator $\wedge$ called a *t-norm*, and OR is replaced by its t-conorm, $\vee$, where $a \vee b = 1 - (1-a) \wedge (1-b)$. Simple examples of t-norms are the minimum of the two argument and the product of its arguments, e.g. with the rule (2):

$$\begin{aligned} &w(x_1, x_2, \ldots) = \\ &\quad m_{I_{11}}(x_1) \wedge (m_{I_{21}}(x_2) \vee m_{I_{22}}(x_2)) \ldots \end{aligned} \tag{3}$$

For the purposes of this work, $\wedge$ =min and $\vee$ =max were used to initially generate the FIS, but the evolutionary algorithm was allowed to mutate these to the product t-norm or the Łukasiewicz t-norm ($x \wedge y = \max(0, x+y-1)$).

For converting the weight value w computed in (3) into the output value $\hat{y}$ of the inference system, one needs to aggregate the weights of all rules with the same consequent (the part following THEN in rule (2)), and then apply a *defuzzification operator*. Aggregation involves taking either the maximum of, or the sum of the weights of all rules with the same consequent. In this work, we used the sum aggregation rule, ie:

$$W_j = \sum_{\{r|c_r=\text{“}y\ IS\ O_j\text{”}\}} w_r(x) \tag{4}$$

where c_r is the consequent of the rth rule, w_r is the weight of the rth rule.

Finally, to produce an inferred output $\hat{y}$, one needs to defuzzify the aggregate weights W_j. A number of possible operators can be employed for this task, but the one we use herein is known as *area defuzzification*. Form new functions

$$\mu_j(x) = \left\{ \begin{array}{cl} m_{O_j}(x) & \text{if } m_{O_j}(x) < W_j \\ W_j & \text{otherwise} \end{array} \right. , \tag{5}$$

as shown in Figure 2. Then the inferred output is the centroid of the area underneath the sum of the μ_j curves:

$$\hat{y} = \frac{\int x \sum_j \mu_j(x) dx}{\int \sum_j \mu_j(x) dx} \tag{6}$$

4.1.1. Fuzzy Logic Source Code

FISPRO is an open source software package available from INRA. implementing fuzzy inference systems, allowing a wide variety of membership functions, different forms of weight computation, aggregation and defuzzification to be specified. It is implemented as a C++ library, with a Java interface provided through JNI that enables an interactive Java program that users can use to design fuzzy inference systems, and experiment with the inference engine.

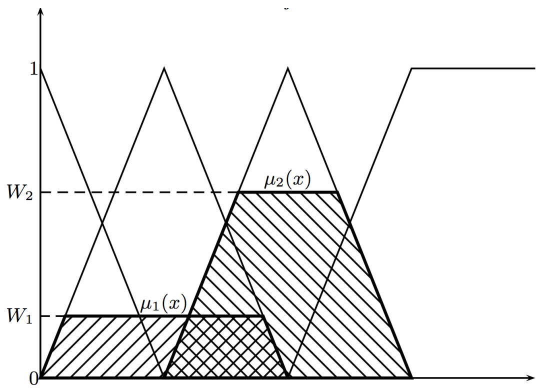


Figure 2: Membership functions

Additionally, FISPRO provides a number of functions for learning rules from training datasets. These were used to initially seed the evolutionary algorithms.

Various performance enhancements were added to FISPRO 3.0 and submitted to the FISPRO maintainers for inclusion into the next release of FISPRO.

4.2. Parametrising the FIS from Real World Data

Since around 1990, people have sought to combine the knowledge representation power of fuzzy inference systems with the learning power of evolutionary algorithms (EA), particularly genetic algorithms. Alander noted some 280 papers have been published on the topic by early 1996 (Alander 1997). A ten year survey by Cordón et al notes the different types of approaches taken to evolving fuzzy inference systems (Cordón, Gomide, Herrera, Homann & Magdalena 2004). Different aspects of the FIS are available to be evolved: the type of FIS (whether Mamdani, or Takagi-Sugeno), the t-norm used in the calculation of the rule weight (3), the rules and their consequents, the number and shapes of membership functions for the inputs and outputs, and the actual parameters of the membership functions. Most commonly, the parameters of the membership functions are evolved, or the rule base is evolved. The evolutionary algorithm used is usually a genetic algorithm (parameters converted into a bitstring representation, which is evolved), although evolutionary strategies (working directly with floating point representations) are also deployed (eg (Cordón & Herrera 1999)), as we do here. When evolving the rule base, individuals of the EA may either be complete FISes (called the Pittsburgh approach) as used in this work, or individual rules (Michigan or Iterative approaches).

In the EvoNF framework (Abraham 2002) all of these aspects can be tuned, but in practice evolving all levels of this framework is computationally prohibitive. Normally, domain knowledge is used to constrain the optimisation search space. We had initially hoped to evolve just the membership function parameters, with the fuzzy rule base being given by domain knowledge. However, the domain knowledge turned out insufficient for the task, so we chose to inform the rule base from the data, by seeding the evolving population using the FPA algorithm (Glorennec 1996), and then further evolving the rule base.

4.3. Representation of the FIS

An evolutionary algorithm requires a representation of the solution, a sequence of evolutionary operators (genetic operators) to generate variation and a selection criterion for removing unsuccessful solutions from the pool.

In this work, we use a direct representation in the form of a list of the parameters for all the membership functions of the input and output fuzzy sets. We also vary which rules are active, and what their consequents are.

For computational efficiency (avoiding the extensive computation of exponentials or other such functions), we use piecewise linear trapezoidal membership functions. This includes triangular and semi-trapezoidal membership functions as a special case. Figure 3 shows the general form of the membership function, and defines the term *core*, where the membership function is 1, and the *support* where the membership function is greater than 0.

4.4. Evolutionary Operators

The operators we implemented were mutation, insertion (splitting), deletion (merging) and crossover, each controlled by a separate parameter.

In the case of mutation, with probability given by the mutation probability parameter `mutConc` (0.01), a number was drawn from $\{0, \ldots, N_c\}$, where N_c is the number of fuzzy sets in the FIS output. If the number drawn was N_c, the rule was toggled between active and inactive, otherwise the rule's conclusion was set to the fuzzy set corresponding the number drawn. Similarly, with probability `mutConj` (0.01), the conjunction operator $\wedge$ was mutated between minimum, product and Łukasiewicz t-norms (which are the t-norms supported by FISPRO).

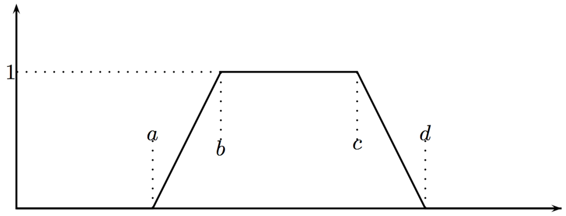


Figure 3: A piecewise linear *trapezoidal* membership function. The area between b and c is known as the *core* and the area between a and d the *support*. If $b = c$, the function is called *triangular*, if $a = b = -\infty$, or $c = d = \infty$ it is known as *infimum* or *supremum semi-trapezoidal* respectively

Mutating membership function parameters is a little more complex. We need to ensure that the fuzzy sets do not expand to engulf neighbouring fuzzy sets, so we require that (a) the support of a membership function does not overlap the core of its neighbours (b) that the the support to overlap the support of its neighbour to ensure complete coverage and (c) that $a \le b \le c \le d$. We also limit the maximum variation to an evolutionary parameter `mutrange` (=0.1). In practice, this means that we calculate a range that maximally satisfies all those constraints, and chose a value randomly from that range. More precisely, if a_i, b_i, etc. represent the parameters of the ith membership function, the core is given by $[b_i, c_i]$ and the support by $[a_i, d_i]$. Let μ=`mutrange`. Then we form:

$$\begin{aligned}
a_i^- &= \max\{a_i - \mu, c_{i-1}\} \\
a_i^+ &= \min\{a_i + \mu, b_i, d_{i-1}\} \\
a_i' &\in [a_i^-, a_i+] \\
b_i^- &= \max\{b_i - \mu, a_i'\} \\
b_i^+ &= \min\{b_i + \mu, c_i\} \\
b_i' &\in [b_i^-, b_i^+] \qquad (7) \\
c_i^- &= \max\{c_i - \mu, b_i'\}
\end{aligned}$$

$$\begin{aligned} c_i^+ &= \min\{c_i + \mu, d_i\} \\ c_i' &\in [c_i^-, c_i^+] \\ d_i^- &= \max\{d_i - \mu, c_i', a_{i+1}\} \\ d_i^+ &= \min\{d_i + \mu, b_{i+1}\} \\ d_i' &\in [d_i^-, d_i^+] \end{aligned}$$

The a_i', b_i', d_i' and d_i' become the new mutated value of the parameters. If any of the $x_i^- > x_i^+$, it is not possible to draw a new value for parameter x_i, so the parameters are left unchanged.

The insertion operator was implemented by replacing a membership function with two new membership functions that split the original core between them. If a, b, c, d are the original function's parameters, the new functions' parameters are:

$$\begin{aligned} a_1 &= a \\ b_1 &= c_1 = b \\ d_1 &= c \\ a_2 &= b \\ b_2 &= c_2 = c \\ d_2 &= d \end{aligned} \tag{8}$$

The resulting membership functions are triangular, but need not remain that way after further mutation.

The deletion operator was implemented as replacing a pair of consecutive membership functions with a single merged function:

$$\begin{aligned} a &= a_1 \\ b &= b_1 \\ c &= c_2 \\ d &= d_2 \end{aligned} \tag{9}$$

These operators were applied with probability `split` (0.01) and `merge` (0.01), which are parameters of the evolutionary algorithm. Because these operations change the number of membership functions describing an input variable, we end up with oddities such as rule bases with 3 genders, so in practice we also specified a Boolean input array to indicate which inputs, and whether the output could have split/merge applied to them. Furthermore, changing the numbers of membership functions invalidates the ruleset. Rather than renumbering the ruleset (which would require resolving the issue of which of the two previous fuzzy sets maps to the new fuzzy set when a merge has happened), we took the approach of reapplying the FPA algorithm to regenerate a new rule base from scratch.

The final evolutionary operator was the crossover operator. This simply selected two parents at random from the pool, and crossed the membership functions for each input with 50% probability, and also crossed rules with matching antecedents with 50% probability (or performed an insertion of a parent 2 rule if its antecedent doesn't exist in parent 1. A later step in the evolutionary algorithm removes identical FISes from the pool.

4.5. Evolutionary Algorithm

A pool of FISes is seeded with a FIS generated using the *Fast Prototyping Algorithm* (FPA) (Glorennec 1996). The FIS pool is iterated over, with the various evolutionary operators described in the previous section applied according to the controlling probabilities. Once the number of FISes in the pool reached `maxPop` (10), selection is applied.

The primary measure of fitness of the FIS is *performance* P, which is the root mean square error of the FIS with respect to the training dataset σ. Some of the items in the training dataset may not match any of the rules well, particularly as the input fuzzy partitions evolve. If the maximum w computed according to equation (3) is less than m (`matchThresh`=0.01), then the item is dropped from the training set. Let $\sigma' = \{j \in \sigma : w(j) \geq m\}$, then performance is calculated from

$$P = \sqrt{\frac{1}{|\sigma'|}\sum_{i\in\sigma'}(\hat{y}_i - y_i)^2}, \tag{10}$$

and the coverage C as

$$C = \frac{|\sigma'|}{|\sigma|}. \tag{11}$$

The advantage to using this is that the FIS can report when its predictive ability is poor, and one can substitute an alternative inference rule (such as random selection from a probability table).

Using performance (eq (10) directly as a fitness function encourages the algorithm to find solutions that diminish coverage. By eliminating difficult to predict conclusions, P can be made arbitrarily small, even zero. If one sets `matchThresh` to zero (ensuring all of the training set is used), then P is dominated by the poorly performing rules, and the evolutionary algorithm has difficulty finding improvements.

An improvement to using P directly is to combine coverage, for instance as a ratio P/C

or as a linear combination $P + \alpha/C$. The former fitness function is particularly prone to finding a FIS that reduces coverage to the point that $P = 0$, which then dominates the evolutionary pool. The latter fitness function has the troublesome α parameter, and the algorithm stagnates once $P \leq \alpha$.

This is a problem of multi-objective optimisation (simultaneously minimising P at the same time as keeping C as high as possible). An alternative approach to multi-objective evolutionary algorithms is Pareto optimisation (Abbass 2006), whereby only Pareto-dominated FIS candidates are eliminated (those for which other FISes in the population are better at both performance and coverage). In practice, this algorithm worked the best of all.

4.6. Implementation and Results

The evolutionary algorithm was coded in C++ as a model running under EcoLab (Standish & Leow 2003), available from the EcoLab website.[2] OpenMP (OpenMP 2002) was used to parallelise the computation of performance and coverage, as well as repopulating the rule base after a a change in the number of fuzzy sets describing the inputs or output. The source code is the NCR.D5 release and relies on the modified version of FISPRO 3.0 (fispro.3.0.D10 see section 4.1.1.) Both are available from the same (EcoLab) website.

he smaller dataset had both advantages and disadvantages but on alance proved the most useful and all the results reported in this paper refer to he product risk timeseries obtained from this dataset. It was further downsized by sampling the customers with a requency 0.01 and of 0.001. This resulted in 285,660 records for the .01 sampling frequency, and 30,627 for the 0.001 sampling frequency. arious rule induction options available within FISPRO were tried, but nly the FPA option could handle such large datasets.

Using a single objective function, P/C could in exceptional cases give good coverage but tended to achieve coverage of only around coverage was around 80–85%.

With the matching threshold parameter was set to 0, coverage is 100% by definition. The evolutionary algorithm still improved the starting ruleset found by FPA, but it doesn't produce as good a solution as the multi-objective methods.

In the multi-objective case the a Pareto front of the best solutions occurs near the edge of the "cliff" where coverage drops off precipitously. The best solution at the end of the run has $P = 3.39 \times 10^{-11}$ and $C = 1.0$.

Extending the dataset to the 0.01 sample rate achieves a performance rate of 0.077 with complete coverage.

Thus optimising just on performance was not nearly as effective as optimising both P and C.

5. NETWORK ANALYSIS

The trust model requires understanding the social networks of clients. Although there is plenty of empirical evidence for small world (Watts 1999) or scale free connectivity (Barabási 2002), there are no fields in the data warehouse which capture the links. Furthermore there are no obvious proxies, e.g. children attending the same school, membership of the same sports clubs might all be harbingers of interactions, but no data of this kind is available. Hence a deeper inference system was needed. The solution is in looking at the time series of investments and determining how one client's investments correlates with another. Correlation between time series is a well understood metric but it can sometimes miss nonlinear interactions commpletely. Mutual information is a more powerful, although computationally more demanding technique (Cellucci, Albano & Rapp 2005).

5.1. Methods

The investment timeseries of the approx 180,000 customers investing in products with known performance data was computed using the technique described in section **??**. Since the absolute balances of the product investment do not carry meaningful information, and even relative balance (monthly investment divided by product balance) is distorted, the timeseries were converted to the range $\{-1, 0, 1\}$, representing withdrawal, no investment and investment respectively in a product. Mutual information was calculated in the usual way between these reduced investment timeseries $I(c, pr, t)$ for customer c, product pr at time t:

$$\begin{aligned} MI(c_1, c_2; \Delta) &= \sum_{x.y} p(x,y) \ln \frac{p(x,y)}{p_1(x)p_2(y)} \\ p(x,y) &\equiv p(I(c_1, pr, t) = x, \\ &\quad I(c_2, pr, t+\Delta) = y) \\ p_1(x) &\equiv p(I(c_1, pr, t) = x) \end{aligned}$$

---

[2] http://ecolab.sourceforge.net

$$\begin{aligned} p_2(y) &\equiv p(I(c_2, pr, t+\Delta) = y) \\ (x,y) &\in \{-1,0,1\}^2 | (x,y) \neq (0,0) \end{aligned} \tag{12}$$

Here the probability distributions were computed by histogramming over the 20 investment products and all timeseries points t (Jan 2002 – Dec 2003). The $(x,y) = (0,0)$ points were excluded because there were a lot of data points where no activity occurred, leading to spurious mutual information. Offsetting the timeseries by Δ in the range $[-3, +3]$ months allows for possible causality to be inferred. By maximising the mutual information over Δ,

$$MI(c_1, c_2) \max_{\Delta} MI(c_1, c_2; \Delta) \tag{13}$$

the sign of Δ for which the mutual information is maximised induces a direction to the network link, pointing from c_1 to c_2 is Δ is positive, and vice versa.

Using the above technique of classifying customers passive investors are largely excluded by the $(x,y) = (0,0)$ exclusion, but regular customers will also tend to add spurious correlation between the timeseries, that is not due to causal influence. Therefore, the dataset was further reduced to just those customers classified as active investors. In the graphs presented with this report, the threshold was taken as a conservative 5%, meaning that only the 120,000 customers classified as active were considered. Further, this dataset was decimated to a final collection of approx 12,000 customers. A mutual information threshold of 1.05 was chosen to capture the most important links between customers. The networks generated in this fashion, illustrated in figure 4 (LGL 2008), typically have around 4000-7000 nodes.

The networks show substantial clustering, as well as restructuring as a function of time. Probably the next phase of research would be to apply standard network metrics such as degree centrality, clustering coefficients and categorising the type of degree distribution.

6. CONCLUSION

This paper identified fuzzy inference systems which could be parametrised against very large real world datasets. A multi-objective evolutionary algorithm significantly improves the performance and coverage of a fuzzy inference system trained on a large dataset over what was obtained by the fast prototype algorithm of FISPRO. This fuzzy system can now be used in the agent based model of client investment behaviour.

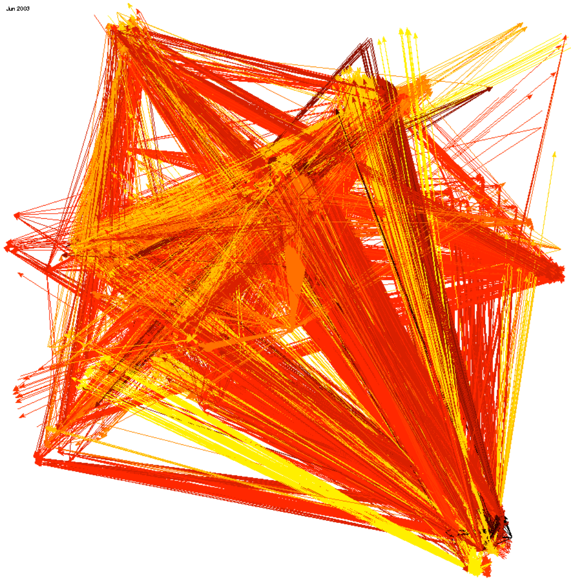


Figure 4: Illustrative client network. using large graph layout techniques to map related nodes close to one another. The client network shows a pronounced hub structure.

ACKNOWLEDGMENTS

This work was supported by grant LP0453657 from the Australian Research Council and a grant of computer time from the Australian Centre for Advanced Computing and Communications.

REFERENCES

Abbass, H. 2006. Pareto-optimal approaches to neuro-ensemble learning, *in* Y. Jin, ed., 'Multi-Objective Machine Learning', Vol. 16 of *Studies in Computational Intelligence*, Springer, Berlin, chapter 18, pp. 407–427.

Abraham, A. 2002. EvoNF: A framework for optimization of fuzzy inference systems using neural network learning and evolutionary computation, *in* 'Proceedings of the 2002 IEEE International Symposium on Intelligent Control', pp. 327–332. arXiv:cs/0405032.

Alander, J. T. 1997. An indexed bibliography of genetic algorithms with fuzzy logic, *in* W. Pedrycz, ed., 'Fuzzy Evolutionary Computation', Kluwer Academic, Boston, pp. 299–318.

Barabási, A.-L. 2002. *Linked*, Perseus, Massachusetts.

Bossomaier, T., Jarratt, D., Anver, M., Thompson, J. & Cooper, J. 2005. Optimisation of client trust by evolutionary learning of financial planning strategies in an agent based model, *in* 'Proc, IEEE Conf. on Evolutionary Computing', pp. 856–863.

Cellucci, C., Albano, A. & Rapp, P. 2005. Statistical validation of mutual information calculations: Comparison of alternative numerical algorithms, *Physical Review E* 71.

Cordón, O., Gomide, F., Herrera, F., Homann, F. & Magdalena, L. 2004. Ten years of genetic fuzzy systems: Current framework and new trends, *Fuzzy Sets and Systems* 141: 5–31.

Cordón, O. & Herrera, F. 1999. A two-stage evolutionary process for designing tsk fuzzy rule-based systems, *IEEE Transactions on Systems, Man, and Cybernetics—Part B: Cybetics* 29: 703–715.

Fulcher, J. 2008. Computational intelligence: an introduction, *Studies in Computational Intelligence* 115: 1.

Glorennec, P.-Y. 1996. Quelques aspects analytiques des systèmes d'inférence floue, *Journal Européen des Systèmes automatisés* 30: 231–254.

LGL 2008. Large graph layout.
URL: *http://bioinformatics.icmb.utexas.edu/lgl/*

OpenMP 2002. *OpenMP C and C++ Application Program Interface*, OpenMP Architecture Review Board. Version 2.0.

Standish, R. K. & Leow, R. 2003. EcoLab: Agent based modeling for C++ programmers, *in* 'Proceedings SwarmFest 2003'. arXiv:cs.MA/0401026.

Watts, D. 1999. *Small Worlds*, Princeton University Press.

Yahoo 2008. Yahoo finance.
URL: *http://www.yahoo.com.au*

ENABLING ADVANCED SIMULATION SCENARIOS WITH NEW SOFTWARE ENGINEERING TECHNIQUES

Judicaël Ribault(a), **Olivier Dalle**(b)

(a)(b)Mascotte project-team, INRIA Sophia Antipolis &
I3S, Université de Nice-Sophia Antipolis, CNRS
B.P. 93, F-06902 Sophia Antipolis Cedex, France.

(a)`Judicael.Ribault@sophia.inria.fr`, (b)`Olivier.Dalle@sophia.inria.fr`

ABSTRACT

In this paper, we introduce new techniques in the field of simulation to help in the process of building advanced simulation scenarios using preexisting simulation components. The first technique consists in using the Aspect Oriented Programming paradigm to capture some of the private data of an existing model component.The second one is an Architecture Description Language (ADL) designed for the Fractal component model, that offers definition overloading and extension mechanisms similar to those found in traditional Object Oriented languages. The benefits of using both techniques are illustrated by simple use cases of network security studies.

1 INTRODUCTION

In the 70's, Zeigler introduced the DEVS formalism (Zeigler 1976): a formalism to represent the hierarchical structure and behavior of discrete-event systems according to the Systems Theory. Later, Zeigler et al. further introduced in their Framework for Modelling & Simulation the concept of Experimental Framework (Zeigler, Kim, and Praehofer 2000). This Experimental Framework separates the computer simulation concerns in two parts: on one hand the model of the System Under Testing (SUT) and on the other hand, the Experimental Frame (EF). Hereafter, we will refer to the part of the Experimental Frame that generates exogenous events (inputs) for the model part, as the scenario part. This approach of separating concerns has benefits, such as allowing a better reusability of components.

From a methodological point of view, reuse allows to: (i) build reference model used in several studies, particularly to compare different solutions and (ii) benefit from user feed-back and/or improvements. Notice there are also situations in which reuse can simply not be avoided. Indeed, we may distinguish two levels of component availability. At source level, reusing an existing code offers enough flexibility to allow any desired modification (but at the cost of losing the results of a previous verification and validation.) On the contrary, when components are only available in compiled object code, reuse necessarily happens without any modification.

Furthermore, the approach of separating concerns may imply some limitations. For example, Systems Theory normally prohibits direct interactions between the scenario part and the inner parts of models, because interactions have first to go through the boundaries of the outer components of the model in order to reach the inner ones. Furthermore, for some studies, it may be useful to extend the previous definition of a scenario to include, in addition to the ability to send exogenous events to the model, the ability of applying structural changes to an existing model (before the simulation starts running).

In this paper, we describe new techniques coming from the field of software engineering that can be used in the field of simulation to get around these limitations while enforcing the separation of concerns principles of the Experimental Framework. Hence, it is worth noting that separating models and scenario allows a better reuse of components in both parts: reuse of a given model with various scenarios, or reuse of a given scenario with various models. In particular, it is often advocated that a model that can be reused multiple times or used in combination with other models can save a many time, money, and human effort (Davis and Anderson 2003).

Section 2 present the software background involved in this paper. Section 3 present the use case in which we present the use of ADL (section 3.2) and AOP (section 3.3).

2 BACKGROUND

This section first introduce the Open Simulation Architecture (OSA) (Dalle 2007), a discrete-event simulator that provides a process-oriented programming model. Then we put the emphasis on the two techniques used for its implementation that are of particular interest for building advanced scenarios.

2.1 Open Simulation Architecture

The goal of OSA is to help users in their simulation activities like building models, developing simulations campaigns, running experiences plans, or analyzing data results. Also, OSA aims at becoming framework for the modelling and simulation community by favoring the integration of new or existing contributions at all levels architecture. Figure 1 represents the OSA architecture. In the left part, the front end-users GUI based on Eclipse framework. In the center part, the functional concerns and in the right part the simulation tasks. Functional concerns resolve one or more typical simulation tasks. Each functional concerns are part of the OSA software components and must be considered optional and replaceable independently from one another. In OSA, handling are almost always hidden in the controller component thus significantly reduce the modelling process, but also simplifies the replacement of any part of the simulation engine. OSA allows to model component-based systems using Fractal component (Bruneton, Coupaye, and Stefani 2004). AOKell, an open implementation in Java of the Fractal component model, provides an aspect-oriented approach to integrate control concerns in component. In practice, the real system is represented by a FractalADL application. This application can then be instrumented using Fractal component capability.

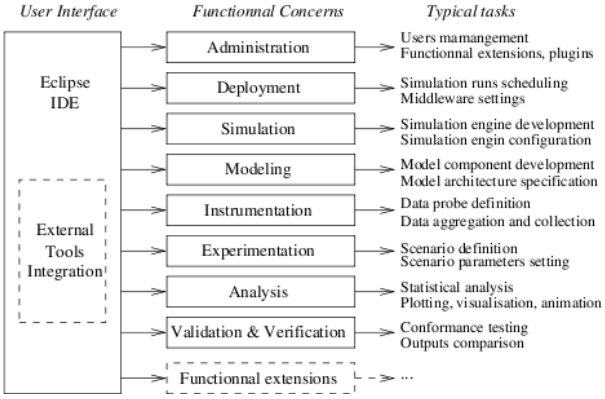

User Interface
Functionnal Concerns
Typical tasks
Eclipse IDE
External Tools Integration
Administration
Deployment
Simulation
Modeling
Instrumentation
Experimentation
Analysis
Validation & Verification
Functionnal extensions
Users mamangement
Functionnal extensions, plugins
Simulation runs scheduling
Middleware settings
Simulation engine development
Simulation engin configuration
Model component development
Model architecture specification
Data probe definition
Data aggregation and collection
Scenario definition
Scenario parameters setting
Statistical analysis
Plotting, visualisation, animation
Conformance testing
Outputs comparison
...

Figure 1: OSA functional architecture.

2.2 Fractal component

Fractal basis development lies in writing components and connections that enable components communication. Fractal specification is based on: (1) hierarchical components that provide a uniform view of applications at different levels of abstraction, (2) shared components that allow modelling and sharing of resources, while preserving hierarchical components, (3) introspection to observe the performance of a system, and (4) (re)configuration capabilities that enable deployment and dynamic system configuration. Furthermore, Fractal is an extensible model because it allows the developer to customize the control capabilities of each application's component. A Fractal component is an unit of deployment that have one or more interfaces. An interface is an entry point to the component. An interface implements an interface type, which specifies the operations supported by the interface. There are two types of interfaces: server interfaces that correspond to the services provided by the component and client interfaces that correspond to services required by the component. A Fractal component is normally composed of two parts: a membrane which possesses functional interfaces and interfaces allowing introspection and (dynamic) configuration of a component, and a content that is made up of a finite set of sub-components.

Figure 2 shows an example of Fractal component. Components are represented by rectangles. The bold line corresponds to the membrane component. The inner part corresponds to the content of the component. Interfaces are represented by round for clients interfaces, and by empty half-round for servers interfaces. Note that internal interfaces allow a hierarchic component to control the exposure of its external interfaces to its sub-components. External interfaces appearing at the top of the components are component control interfaces. dashed line represent connections among components. Fractal provide a Architecture Description Language (ADL) (Clements 1996; Medvidovic and Taylor 2000) to describe applications architecture.

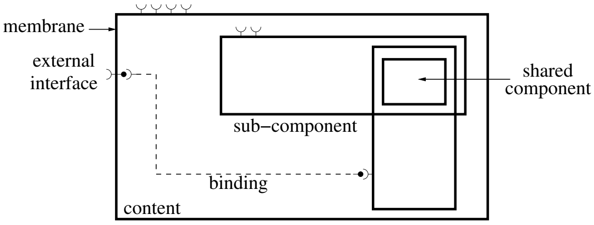


Figure 2: Fractal component example.

2.2.1 Fractal ADL

FractalADL is a XML language to describe the architecture of a Fractal application: components topology (or hierarchy), relationship between client and server, name and initial value of components attributes. A FractalADL definition can be divided into several subs definitions and several files. Moreover, the language supports a mechanism to ease the extension and redefinition through inheritance. The motivation for such scalability is twofold. On the other hand, the component model itself is extensible, it is possible to attach an arbitrary number of components controllers. There are multiple uses for a given ADL definition: deployment, verification, analysis, and so on. FractalADL allows to separate concerns because model definition can be split in multiple files. ADL language is interpreted by a specialized component of Fractal called a Factory: to read completely (recursively) a description of a Fractal application, just send a request to the Fractal Factory to read and instantiate the root component of the application. To instantiate the various components, the factory creates a Fractal Abstract Syntax Tree (AST), where each node corresponds to a XML entity of the ADL.

2.3 Aspect-Oriented Programming

Aspect-Oriented Programming (AOP) (Elrad, Filman, and Bader 2001) is a new paradigm for modularizing applications with many concerns. AOP goals are (i) separation of concerns: the goal is to design systems so that functions can work independently of other functions, and so it is easier to understand, design and manage complex interdependent systems; (2) crosscutting interactions: it is not easy to modularized common-interest concerns used by several modules, like logging service; (3) dependencies inversion: instead module use well-known services, the well-know service shall use modules.

3 ADVANCED SCENARIOS CASE STUDIES

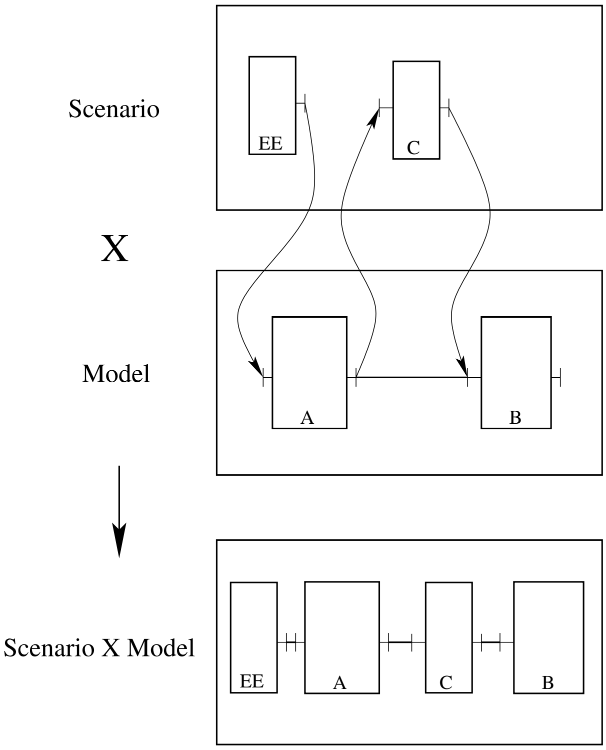


Figure 3: Reuse and adapt a model of reference.

We present hereafter two case studies to illustrate the new techniques we propose to build advanced scenarios reusing existing component models that are only available in compiled form, at execution level (for example because it came after a long validation and verification process, or because we want to keep the source code secret). Figure 3 show the composition of the complex scenario and the reference model. The reference model contains two components A and B. The complex scenario adds a new component C between A and B, and a new component EE which generates exogenous events. The composition is the result of the model and the scenario. In order to build such a composition we propose to use (i) an Architecture Description Language (ADL) with overloading capability like FractalADL and (ii) Aspect-Oriented Programming (AOP) like AspectJ. to extend the reusability of a model through a simple case study To illustrate this kind of composition we build a practical example: a small security case study based on a reference model in which a user establishes an FTP session with a server using the unsecured version of the protocol. The case study will consist in simulating a Man-In-The-Middle attack (MITM) and a Spy-ware version of the client.

3.1 Case study

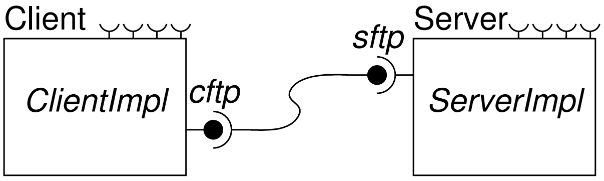


Figure 4: Components layout of File Transfers Protocol case study.

As described previously, we propose to use AOP and ADL in an original way to override difficulties in reusing models. We choose in this paper to show the cost and benefits through a simple case study. First, let us assume that we have a model we want to reuse to test different security flaws. There is a model representing the Basic operation of a server File Transfer Protocol (FTP). This simple model has not been developed in order to be used in this study, we are not supposed to have the source code, and even we need to test the safety of this protocol. Figure 4 shows the architecture of the model, and Listing 1 details its implementation in FractalADL. Line 4 specifies the name of this model, line 6-11 correspond to the client definition and line 12-19 to the server definition. Line 7-9 and 14-16 describe client and server interfaces used by the binding on line 20-21.

Listing 1 Fractal ADL definition used to implement layout of figure 4.

```
01<?xml version="1.0" encoding="ISO-8859-1" ?>
02<!DOCTYPE definition skipped ... >
03
04<definition name="ftp">
05
06  <component name="Client">
07    <interface name="cftp"
08                     role="client"
09                     signature="FTPService"/>
10    <content class="ClientImpl"/>
11  </component>
12
12  <component name="Server">
14    <interface name="sftp"
15                     role="server"
16                     signature="FTPService"/>
17    <content class="ServerImpl"/>
18  </component>
19
20  <binding client="Client.cftp"
21                 server="Server.sftp"/>
22</definition>
```

The protocol represented by this model is a two-party protocol. We will denote the two parties by the name Client and Server (Client want to be authenticated on Server). The model works like this : the client send the users login and password to the server to be authenticated. To do this, client ask his interface (cftp, declared line 07) to obtain connection with the server. In this study, we focus on the login process to test security flaw.

From this model, we propose a new reusing approach. First, we will show how to add a man in the middle attacker in this model using the overload capability of FractalADL. Second, we will show how to simulate spy-ware on client using the overload capability of FractalADL and AOP.

3.2 Man-in-the-middle attacker with Fractal ADL

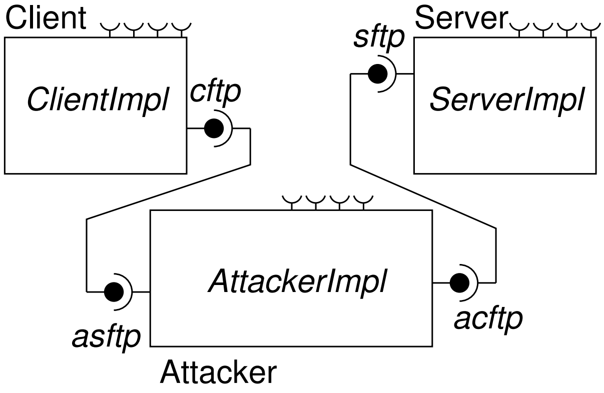


Figure 5: Components layout of Fractal's MITM attack.

From the original model describe in section 3.1, we want to test the ftp login process security. We decide to test the security against a man-in-the-middle attacker. In the man-in-the-middle setting (MITM), there is a third party called Adversary. All the communication between Client and Server are intercepted by Adversary. Thus both Client and Server talk to Adversary and cannot communicate directly with each other. Adversary need to transmit information between Client and Server, but - it's the security break - he can read, change, or drop transmit depending on his settings.

What makes this case interesting is to modify the original FTP topology (figure 4) to obtain the new topology describe in figure 5. In practice, we need to add a new component inside a model. Like in reality, Adversary need to mimic Server interface and Client Interface. In fact, Adversary need to imitates Server for the Client, and imitates Client for the Server. Figure 5 show the new architecture we want to obtain compared to figure 4 section 3.1. Since model is locked, we cannot change his topology directly in source code. Listing 2 shows how to use the FractalADL overload capability to overload the topology. Line 04 show we extend the original ftp model in a new model called mitm-ftp. Line 06-14 represent the declaration of the new Adversary component. And line 16-19 demonstrate how overload the original binding between Client and Server by a new binding between Client and Adversary, and between Adversary and Server. With this topology, communication between the Client and the Server pass through the Adversary.

Listing 2 Fractal ADL definition used to implement layout of figure 5.

```
01<?xml version="1.0" encoding="ISO-8859-1" ?>
02<!DOCTYPE definition skipped ... >
03
04<definition name="mitm-ftp" extends="ftp">
05
06 <component name="Adversary">
07   <interface name="acftp"
08                  role="client"
09                  signature="FTPService"/>
10   <interface name="asftp"
11                  role="server"
12                  signature="FTPService"/>
13   <content class="AdversaryImpl"/>
14 </component>
15
16 <binding client="Client.cftp"
17               server="Adversary.asftp"/>
18 <binding client="Adversary.acftp"
19               server="Server.sftp"/>
20</definition>
```

This example shows how to modify a model to include new component or change topology. The overload capability of Fractal ADL permit to reuse and change some specification of the model like topology. In fact, in our example, communication between the Client and the Server go through the Adversary but the FTP model have not been modified. We build a new model extending the original FTP model, and overload the binding between the Client and the Server. In the next section, we use FractalADL to add a new component and change the topology, but we also demonstrate how to use AOP. The next section described the FTP model with a spy-ware inside the client.

3.3 Spy-ware with aspect-oriented programming

In this section, we demonstrate how using Fractal ADL and aspect-oriented programming we can add a spy-ware (Stafford and Urbaczewski 2004) into the Client from the original FTP model. Spy-ware is the name given to the class of software that is surreptitiously installed on a computer and monitors users activities and reports back to a third party on that behavior [Anon, 2004; Daniels, 2004; Doyle, 2003; Taylor, 2002]. We want to model a spy-ware inside the Client of the FTP model. The goal of this attack is to take the user login and password when typed in. Spy-ware send all information to a third party using the network. The model architecture we want to obtain is shown in figure 6. We see the Client is connected to a third entity (Spy) and contain a Spy-Ware inside his implementation.

Listing 3 shows a solution using Fractal ADL and AOP to introduce spy-ware in original FTP model. Using the extension capability of Fractal ADL, we add a new spy interface to the Client component, we add a Spy component and we bind the Client and

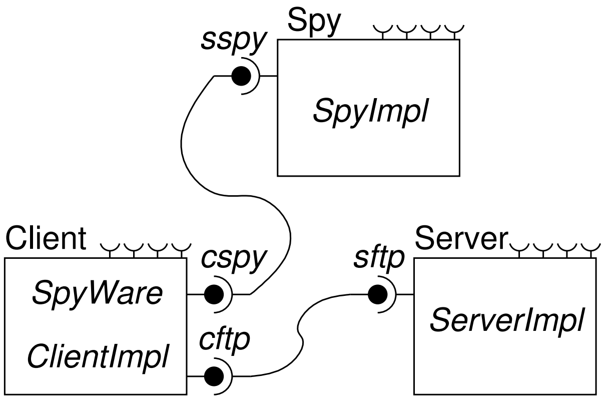


Figure 6: FTP model with Spy-Ware in Client.

the Spy together. Line 04 show how to create a new model extending the original FTP model. Line 06-17 represent the Spy component, line 07-09 represent the interface for connecting with the Spy component. Line 13-17 represent the Client component declared in the original FTP model, line 14-16 show the new interface added to the Client component. Line 19-20 represent the binding to connect the Client with the Spy component.

Listing 3 Fractal ADL used to implement layout of figure 6.

```
01<?xml version="1.0" encoding="ISO-8859-1" ?>
02<!DOCTYPE definition skipped ... >
03
04<definition name="spyware-ftp" extends="ftp">
05
06  <component name="Spy">
07   <interface name="sspy"
08                   role="server"
09                   signature="SpyService"/>
10   <content class="SpyImpl"/>
11  </component>
12
13  <component name="Client">
14   <interface name="cspy"
15                   role="client"
16                   signature="SpyService"/>
17  </component>
18
19  <binding client="Client.cspy"
20                server="Spy.sspy"/>
21</definition>
```

AOP allows us to introduce new code into objects without the objects is needing to have any knowledge of that introduction. The FTP model has been validated and we don't have the source code so we can't change it to introduce some concerns about spy-ware. The Listing 4 show how using AOP we can add some concerns inside a model.

Line 01 explain we want to intercept a method call, and do something before the method was called. Line 02 show the method we want to intercept, it's all methods from the FTPService java interface called by a ClientImpl class. Line 03 add a condition, only component bound with a Spy component are concerned. Line 05 ask the Client interface connected to the Spy component to have this one. Line 06 call through the connection with the Spy the send method to send data. This aspect (written in AspectJ) represent the Spyware, the Spy component represent the third party waiting for data to analyze.

Listing 4 Fractal ADL used to implement layout of figure 6.

```
01 before(ClientImpl b) :
02   call(* FTPService.*(..)) && this(b)
03   && if(isBinding(b)) {
04   try {
05     SpyService spyS = b.lookupFc("cspy");
06     spyS.send(thisJoinPoint.getArgs()[0]+"");
07   } catch (NoSuchInterfaceException nsie) {
08     ...
09   }
10 }
```

This example shows how to modify a model to include new component, change topology and instrument a component. The capability of AOP to inject some code inside the model allow to read variables of the model. Here we demonstrate how a third component can access the login and password field during the login process of the client on server.

4 CONCLUSION

We have shown how ADL and AOP techniques can be used to extend the reusability of a model. Both techniques offer new ways to create a complex scenario without modifying the original model. Hence, the model remains valid which saves additional costs and efforts. The ADL allows to build a composition of the model with the scenario by overloading some model definitions like bindings. AOP helps to add some code into the model, for example to allow a third party component to access an existing model's private data. However, this latter technique must be used with extreme care in order to guarantee that the code newly inserted in a component will not change its behavior. However, tools can be built to make automatic verifications on the code inserted and ensure this non-interference property. Our planned future works are to build a new DEVS-compliant engine for OSA in order to experiment these techniques on existing DEVS models. However, we want also to further investigate the benefits and drawbacks of using ADL and AOP techniques with the DEVS Modelling & Simulation framework. A last direction we want to explore is the identification of practical use cases in which such techniques prove to be useful, in particular in the networking and security area, where models and scenarios exhibit a priori a high complexity(Seo 2006).

ACKNOWLEDGMENTS

This work is co-supported by the European IST-FET "AEOLUS" project and the French National Research Agency (ANR) "SPREADS" and "OSERA" projects.

REFERENCES

Bruneton, E., Coupaye, T., and Stefani, J. B.. 2004, February. The fractal component model specification. Available from http://fractal.objectweb.org/specification/. Draft version 2.0-3.

Clements, P. C. 1996. A survey of architecture description languages. 16: IEEE Computer Society.

Dalle, O. 2007, Februray. Component-based discrete event simulation using the Fractal component model. In *AI, Simulation and Planning in High Autonomy Systems (AIS)-Conceptual Modeling and Simulation (CMS) Joint Conference*. Buenos Aires, AR.

Davis, K. P., and Anderson, A. R.. 2003. *Improving the composability of department of defense models and simulations*. RAND Technical report available at http://www.rand.org/publications/MG/MG101/ (last accessed April 2008).

Elrad, T., Filman, R. E., and Bader, A.. 2001. Aspect-oriented programming: Introduction. *Commun. ACM* 44 (10): 29–32.

Medvidovic, N., and Taylor, R. N.. 2000. A classification and comparison framework for software architecture description languages. *IEEE Trans. Softw. Eng.* 26:70–93.

Seo, H. S. 2006, July. Network security agent DEVS simulation modeling. *Simulation Modelling Practice and Theory* 14 (5). doi:10.1016/j.simpat.2005.08.010.

Stafford, T. F., and Urbaczewski, A.. 2004. Spyware: The ghost in the machine. 291–306: Commun. AIS 14.

Zeigler, B. P. 1976. *Theory of Modelling and Simulation*. Wiley & Sons, NY.

Zeigler, B. P., Kim, T. G., and Praehofer, H.. 2000. *Theory of Modeling and Simulation*. Academic Press, Inc.

AUTHOR BIOGRAPHIES

JUDICAEL RIBAULT received his M.Eng. from University of Nice-Sophia Antipolis in 2007. In 2007, he joined the MASCOTTE common project-team of the I3S-UNS/CNRS Laboratory and INRIA, in Sophia Antipolis, where he started to work on the OSA project as a Software Engineer. In 2008, he started a PhD in the MASCOTTE team, in the context of the ANR SPREADS project, investigating on new Component-based Software Engineering techniques for large scale Peer-to-peer systems simulation.

OLIVIER DALLE is associate professor in the C.S. dept. of Faculty of Sciences at University of Nice-Sophia Antipolis (UNS). He received his BS from U. of Bordeaux 1 and his M.Sc. and Ph.D. from UNS. From 1999 to 2000 he was a post-doctoral fellow at the the French space agency center in Toulouse (CNES-CST), where he started working on component-based discrete event simulation of complex telecommunication systems. In 2000, he joined the MASCOTTE common project-team of the I3S-UNS/CNRS Laboratory and INRIA, in Sophia Antipolis.

METAMODELLING FOR ANALYZING SCENARIOS OF URBAN CRISIS AND AREA STABILIZATION BY APPLYING INTELLIGENT AGENTS

Agostino Bruzzone[a], **Achille Scavotti** [b], **Marina Massei** [c], **Alberto Tremori**[d]

[a] [b] McLeod Institute of Simulation Science
DIPTEM University of Genoa
Via Opera Pia, 15, Genoa, 16145, ITALY

[c]Liophant Simulation

[d] MAST s.r.l
Piazza Lerda, 1, Genoa, 16158, ITALY

[a] agostino@itim.unige.it, [c]marina.massei@liophant.org, [d]alberto.tremori@mastsrl.eu

ABSTRACT

The paper proposes an experimental analysis on a civil disorder scenario where innovative Computer Generated Forces (CGF) are tested. In paper is provided a description of the experimental design made for the definition of the metamodels to be applied in two different cases: urban demonstration and riots as well as the subsequent stabilization period; the two cases related to the same urban scenario are tested in experimental campaigns. In this case metamodels are developed by regression equations in each scenario. So are defined the different metamodels (5 per campaign) and from each one it is possible to apply response surface methodology (RSM) for target function analysis considering multiple independent variables.

Keywords: metamodelling, intelligent agents, urban crisis

1. INTRODUCTION

Metamodels can be used effectively in order to conduct analysis on the experimental results without request to run the simulator; so optimization analysis, quick investigation of specific areas, general criteria.

In a complex scenario involving CGF that introduce many human factors as well as very complex relations, it becomes critical to develop procedures to proper analyze a very wide spectrum of alternatives; the course of action in this context can results very different and in order to support the user the metamodelling approach could be pretty interesting; for these reason in the following the metamodelling analysis is proposed with a set of possible solutions demonstrating the use of this classical techniques in Scenario analysis where Intelligent CGF are used.

Considering the high number of parameters and configuration there are many possible sub-analysis; in the proposed case this is limited to two different configuration of the scenario, each one to be evaluated in term of multiple target functions and input variables; the set proposed is based on 5 input and 5 output specific for each configuration just as example concentrated on some of more interesting and significant parameters.

This paper is based on experimental results obtained using CGF developed in PIOVRA project (Poly-functional Intelligent Operational Virtual Reality Agents); this research was devoted to test an innovative Federation integrating wargaming systems and intelligent CGF and to analyze the potential of their combined used on complex scenarios. The techniques used for these test include Data Analysis, Design of Experiment, Simulations Runs, Statistical Techniques; in fact the results proposed in this paper has been obtained by applying ANOVA and Mean Square pure Error for estimating the impact of stochastic factors.

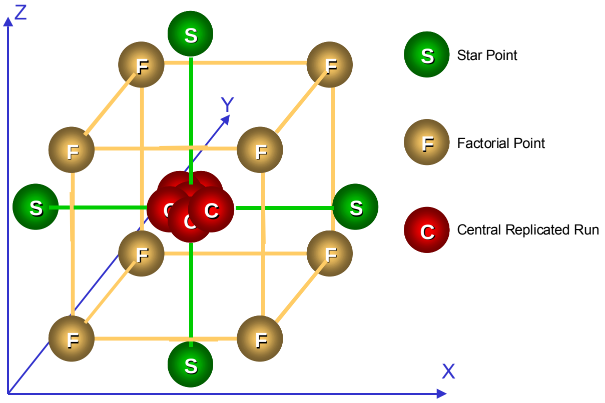


Figure 1: CCD Example for 3 Variable Civil Disorder Metamodelling

A detailed description of the reference input data and experimental Scenario is defined in PIOVRA project as well as all the metrics for measuring the performances and capabilities of agents.

2. CGF FOR URBAN DISORDERS

The present research is related to the necessity to test new intelligent CGF on complex scenarios; these CGF are able to develop autonomously proper reactions and response to the scenario evolution considering their individual perceptions, organizational structures, specific profiles as well as human behavior modifiers dynamically evolving during the simulation based on events and activities.

These CGF represents an important step forward for complex scenario analysis and in particular their major benefits emerge when the unit to model are characterized by complex behavior.

A very good example in this sector is proposed by the urban disorders, where many not military entities are involved; these entities have not clear, predefined, rules and react dynamically to the scenario evolution; for instance demonstration growth and decrease based on all the events and actions reproduced by each single entity: police forces, single agitators, terrorists, political movements, criminal organizations, militia and military forces, etc In this case the CGF was developed by DIPTEM for detailed simulation reproducing from large crowd to small groups and even single individuals; the simulation is characterized by a combined (discrete and continuous combined models) and stochastic modelling; however in a such complex scenario it becomes critical to properly tune the parameters defining behavior of different objects as well as characterizing the scenario; for this reason it was completed a metamodelling devoted to identify the correlation among the scenario independent variables and target functions (i.e. maximum violence level reach during the demonstration).

In this case two different experimental campaign was defined to reproduce two different configuration of the scenario from user point of view:

- Campaign 1: Short Term evolution of the civil disorders; this is devoted to investigate how the crisis evolves, so it is articulated over a timeframe of about 24 hours and cover just civil crisis start up, evolution and its end.
- Campaign 2: Medium Term evolution of the urban disorders; this is devoted to investigate how the after crisis moves to a stabilization of the urban areas and covers between 3-15 days after a big demonstration/riot endings.

For each campaign it was defined a set of variable sot be used example for testing the effectiveness of developing metamodels for this context; at the same example of metamodels are proposed for the specific scenario developed that covers a medium town size (about 300'000 inhabitants) where three different ethnic groups operate; the town police force is divided in two organization (federal and local) with different attitude respect local population and military forces are present to protect strategic infrastructures (i.e. embassies); on the ground two criminal organization are controlling specific areas of the town and also the local health care system is modeled.

The results obtained for this scenario in the two different configuration are specific, but the proposed methodology is pretty general.

3. METAMODELLING

From the computational perspective, the concept of metamodeling is used in mathematics, and is practically applied in computer science and computer engineering/software engineering.

In computer science and related disciplines, metamodeling is the construction of a collection of "concepts" (things, terms, etc.) within a certain domain. A model is an abstraction of phenomena in the real world, and a metamodel is yet another abstraction, highlighting properties of the model itself.

This model is said to conform to its metamodel like a program conforms to the grammar of the programming language in which it is written. In this case we can obtain metamodels by calculating the regression in various points of the domain of the chosen variables.

Response surface methodology (RSM) explores the relationships between several explanatory variables and one or more response variables. The method was introduced by G. E. P. Box and K. B. Wilson in 1951. The main idea of RSM is to use a set of designed experiments to obtain an optimal response. Box and Wilson suggest using a first-degree polynomial model to do this. They acknowledge that this model is only an approximation, but use it because such a model is easy to estimate and apply, even when little is known about the process.

An easy way to estimate a first-degree polynomial model is to use a factorial experiment or a fractional factorial designs. This is sufficient to determine which explanatory variables have an impact on the response variable(s) of interest. Once it is suspected that only significant explanatory variables are left, then a more complicated design, such as a central composite design can be implemented to estimate a second-degree polynomial model, which is still only an approximation at best. However, the second-degree model can be used to optimize (maximize, minimize, or attain a specific target for) a response.

Some extensions of response surface methodology deal with the multiple response problem. Multiple response variables create difficulty because what is optimal for one response may not be “very” optimal for other responses. Other extensions are used to reduce variability in a single response while targeting a specific value, or attaining a near maximum or minimum while preventing variability in that response from getting too large.

Significant criticisms of RSM include the fact that the optimization is almost always done with a model for which the coefficients are estimated, not known. That is, an optimum value may only look optimal, but be far from the truth because of variability in the coefficients.

A contour plot is frequently used to find the responses of two variables to find these coefficients by including a large number of trials in each and combinations of them, and using some sort of interpolation to find potentially better intermediate values between them. But since experimental runs often cost a lot of time and money, it can also be difficult to pinpoint the ideal coefficients, as well; there are frequently strategies used to find those values with minimal runs.

4. EXPERIMENTAL DESIGN

In this case the metamodels are obtained by applying regression techniques; in the proposed case the related equations are 2nd order polynomial approximation of the target functions in function of the five variables considered. The experimental campaign has given the elements to identify the metamodels through design of experiments (DOE).

In this case it is applied a design of experiments based on central composite design (CCD) that is combining a factorial design with central replicated runs for variance estimation and star point for metamodel stabilization as proposed in the above figure for a 3 independent variable case; in our example the CCD involves a set of runs composed as following.

To support the analysis the authors developed a simple analysis tool in VBA© in order to process all the data and obtain response surfaces on request by specifying which input and output to consider.

4.1. Direct civil disorder: campaign 1

As anticipated this configuration of the experimental scenario is devoted to reproduce the urban disorders on short term, concentrating on its evolution, related counter actions by police, militia and military forces as well as contingencies and concurrent actions carried out by different actors (i.e. criminal activities taking opportunities of riots).

Coolness parameter represent the profile of the units in term of capability to resist to provocation, while terrorist actions is a discrete variable representing the following alternatives:

- no terrorist activity at all;
- limited terrorist activity to a single action concurrent with the civil disorders;
- multiple terrorist actions coordinated with civil disorders.

The variables of Campaign 1 are:

- X1: N° Demonstrators;
- X2: Coolness Military;
- X3: N° Policemen;
- X4: Coolness Police;
- X5: Terrorist.

Regressions, campaign 1:

- Campaign1, target function 1 – Overall Importance of Demonstrator Goal Achieved

$$Y_{1,1} = 6364.38 - 0.25x_1 - 145.51x_2 - 3.76x_3 - 77.58x_4\,391.68x_5 + 0.00004x_1^2 + 1.13x_2^2 + 0.01x_3^2 + 1.08x_4^2 + 150.17x_5^2 \qquad (1)$$

- Campaign1, target function 2 –Value of Losses on critical entities

$$Y_{1,2} = 732.54 - 0.01x_1 - 13.97x_2 - 0.24x_3 - 8.33x_4 - 17.93x_5 + 0.000002x_1^2 + 0.09x_2^2 + 0.0004x_3^2 + 0.09x_4^2 + 47.05x_5^2 \qquad (2)$$

- Campaign1, target function 3 - Maximum Level of violence during the Civil Disorders

$$Y_{1,3} = 3742.38 - 0.18x_1 - 80.20x_2 - 2.49x_3 - 43.03x_4 + 457.22x_5 + 0.00003x_1^2\,0.60x_2^2 + 0.003x_3^2 + 0.61x_4^2 + 33.88x_5^2 \qquad (3)$$

- Campaign1, target function 4 – Average Level of violence during the Civil Disorders

$$Y_{1,4} = 210365.5 - 8.01x_1 - 4877x_2 - 43.94x_3 - 1307.16x_4 + 10173.87x_5 + 0.001x_1^2 + 33.99x_2^2 + 0.07x_3^2 + 15.23x_4^2 + 1918.45x_5^2 \qquad (4)$$

- Campaign1, target function 5 - Total Number of Disable People within Military Forces

$$Y_{1,5} = 801.03 - 0.02x_1 - 16.93x_2 - 0.41x_3 - 9.50x_4 + 55.56x_5 + 4.42E-06x_1^2 + 0.126x_2^2 + 0.0005x_3^2 + 0.12x_4^2 + 11.18x_5^2 \qquad (5)$$

4.2. Civil disorder stabilization: campaign 2

In this case the experimentation of scenario is focusing on reproduction of medium term evolution, considering the stabilization of the area after significant urban disorders and includes the influence of population, ethnic groups, police and para-militar forces, military units, terrorists, criminal organizations, etc.
The variables of Campaign 2 are:

- X1: N° Military Units;
- X2: Coolness Military;
- X3: N° Police;
- X4: Gang’s Force;
- X5: Terrorist.

Regressions, campaign 2:

- Campaign2, target function 1 - Overall Importance of Demonstrator Goal Achieved

$$Y_{2,1} = -730.088+131.773x_1 + 14.241x_2 + 1.343x_3 - 1.175x_4 - 138.492x_5 - 16.126x_1^2 - 0.095x_2^2 - 0.002x_3^2 + 0.003x_4^2 + 61.996x_5^2 \quad (6)$$

- Campaign2, target function 2 - Value of Losses on critical entities

$$Y_{2,2} = 0 + 0x_1 + 0x_2 + 0x_3 + 0x_4 + 0x_5 + 0x_1^2 + 0x_2^2 + 0x_3^2 + 0x_4^2 + 0x_5^2 \quad (7)$$

- Campaign2, target function 3 - Maximum Level of violence during the Scenario Evolution

$$Y_{2,3} = 3850.72 - 3.58x_1 - 115.61x_2 + 1.04x_3 - 0.76x_4 + 37.83x_5 + 2.22x_1^2 + 0.85x_2^2 + 0.001x_3^2 + 0.001x_4^2 + 0.53x_5^2 \quad (8)$$

- Campaign2, target function 4 - Average Level of violence during the Scenario Evolution

$$Y_{2,4} = 329353.3 + 4328.04x_1 - 11109.3x_2 + 119.24x_3 + 80.38x_4 - 8409.71x_5 - 456.18x_1^2 + 79.33x_2^2 - 0.11x_3^2 + 0.15x_4^2 + 11420.19x_5^2 \quad (9)$$

- Campaign2, target function 5 - Total Number of Disable People within Military Forces

$$Y_{2,5} = 34.2428 + 19.1784x_1 - 1.2413x_2 + 0.0567x_3 - 0.1267x_4 - 13.3951x_5 - 2.3918x_1^2 + 0.0093x_2^2 - 0.0001x_3^2 + 0.0003x_4^2 + 5.9329x_5^2 \quad (10)$$

5. IDENTIFICATION OF METAMODELS

Experimental designs used in RSM must make trade-offs between reducing variability and reducing the negative impact that can be caused by bias. In the case of the proposed civil disorder experimental campaign the authors have selected some of the most interesting response surface, in which some minimum or maximum values are identified by a particular condition of the input variables.

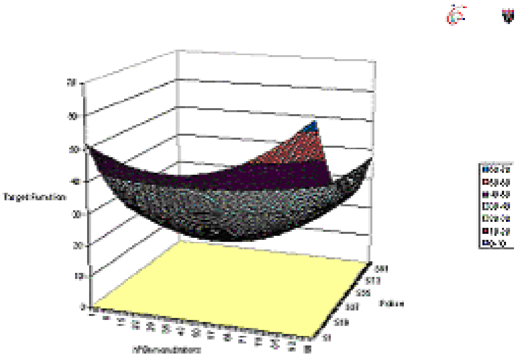


Figure 2: Campaign I - Total number of Disable People within our forces

The present graph propose the estimation of total number of Disable People respect the number of Demonstrators [the axis is from 0 to 99 corresponding to the range of analysis 2500-4500] and the number of policeman [250 to 550 policemen]; the RSM demonstrates that there is an ideal correlation between number of police forces and demonstration size if all the other variables are fixed on reference values; this is a simple 3D projection of the Metamodel that in reality is depending from 5 variables plus all the other boundary conditions (i.e. weather, population attitudes etc.). This is an example of the proposed analysis.

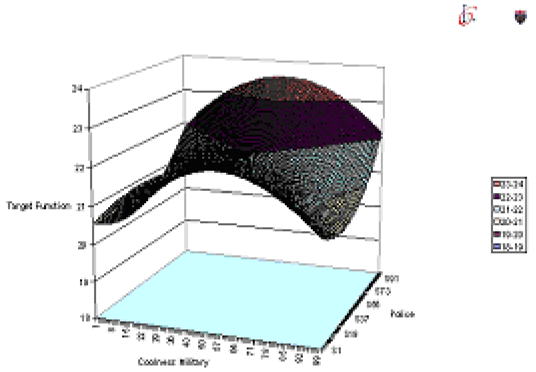


Figure 3: Campaign II - Total number of Disable People within our forces

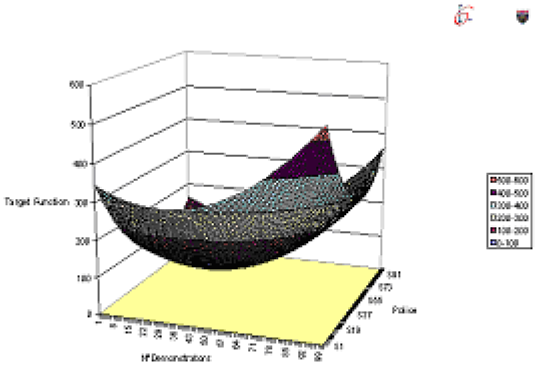


Figure 4: Campaign I - Importance factor for demonstrator goal achieved

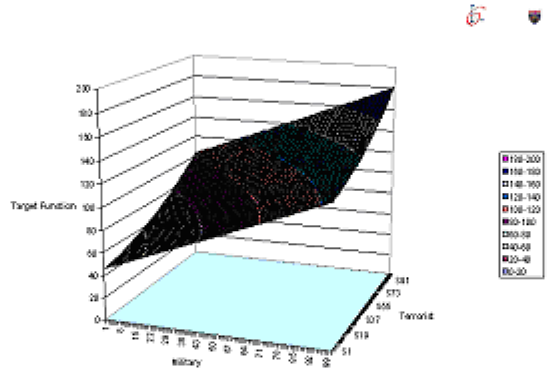


Figure 5: Campaign II - Maximum level of violence

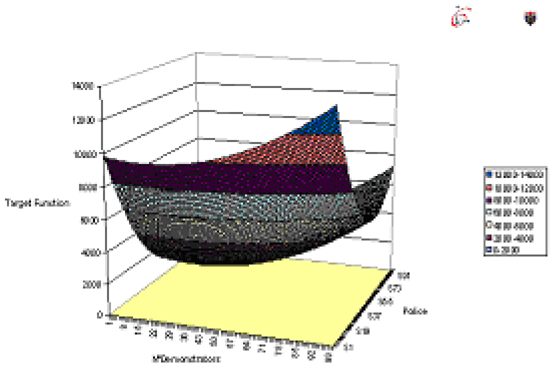


Figure 6: Campaign I - Average level of violence

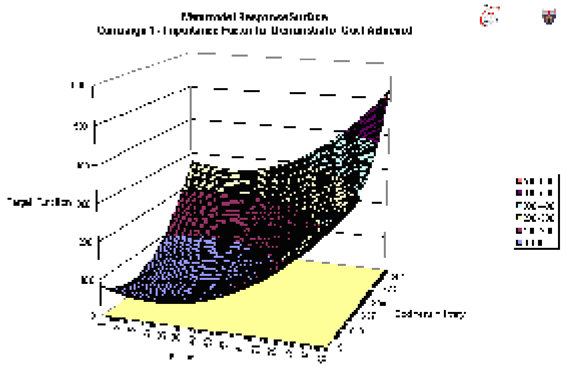


Figure 7: Campaign II - Importance Factor of Goal Achieved

6. CONCLUSIONS

The proposed analysis confirmed the effectiveness of PIOVRA Agents and the possibility to develop innovative Computer Generated Forces to investigate complex scenarios; the experimental analysis performed is an useful guideline for process definition to study the influence of the different factors on the target functions for supporting further application of Scenario analysis and CGF evaluation. The civil disorder experimental analysis allowed to demonstrate the importance of design of experiments and metamodelling in supporting course of action mapping in complex interoperable simulation combining intelligent CGF and different systems.

In addition the authors completed other additional tests on the Interoperability based on PIOVRA Agents with others Simulators has been conducted in HLA; it has been completed the demonstration of integrating in a HLA Federation PIOVRA, JTLS (a constructive simulator by Roland and Associates) and COCODRIS (a Virtual Simulator developed by DIPTEM) The combination of Theater Wargaming, Intelligent CGF and Synthetic Environment in HLA provides additional opportunities for applying experimental analysis methodologies and it confirms the potential to expand the possibilities in term of representation, understanding and control of the battlefield as expected in the present research. Further tests have been conducted with others solutions (i.e. Sheharazade from University of Arizona for Stability and Support Operations).

ACKNOWLEDGMENTS

The authors thanks Italian and French Ministry of Defense (MoD) and EDA (European Defense Agency) for their support; a special thanks to the LSIS Team that was actively involved in the project.

REFERENCES

Amico Vince, Guha R., Bruzzone A.G. (2000) "Critical Issues in Simulation", *Proceedings of Summer Computer Simulation Conference*, Vancouver, July

Bruzzone A.G., Massei M., B.M. (2007) "Demonstration for Human Behavior Modeling Within Civil Disorder Scenarios", *Proceedings of SCSC2007*, San Diego CA, July

Bruzzone A.G., Rocca A. B.M., (2006) "Movements and Attrition within Areas affected by Civil Disorders", *Proceedings of EMSS*, Barcelona, October

Bocca E., Bruzzone A.G., Rocca A. (2006) "Algorithms devoted to Reproduce Human Modifiers in Polyfunctional Agents", *Proceedings of SCSC*, Calgary, August

Bruzzone A.G., Kerckhoffs (1996) "*Simulation in Industry*", Genoa, Italy, October, Vol. I & II, ISBN 1-56555-099-4

Bruzzone A.G., Giribone P. (1998) "Decision-Support Systems and Simulation for Logistics: Moving Forward for a Distributed, Real-Time, Interactive Simulation Environment", *Proc. of the Annual Simulation Symposium IEEE,* Boston

Bruzzone A.G., Petrova P., B.M., B.C. (2004) "*Poly-functional Intelligent Agents for Computer Generated Forces*", Proceedings of Wintersim2004, Washington DC, December

Bruzzone A.G., Williams E.(2005) "*Summer Computer Simulation Conference*", SCS International, San Diego

Fishwick, P. (1995) "*Simulation Model Design and Execution*", Prentice Hall, New Jersey

Hamilton John A. Jr., Nash David A., Pooch Udo W. (1996) "*Distributed Simulation*", CRC Press

Kuhl F., Weatherly R., Dahmann J., Kuhl F., Jones Anita, (1999) "*HLA: Creating Computer*

Simulation Systems", Prentice Hall; ISBN: 0130225118

Rozenblit J., Bruzzone A.G., Momem F., Saetta S. (2007) "A Modelling and Simulation of Coarse of Action (COA) Consequences in Stability and Support Operations", *Proc.of EUROSIW2007*, Santa Margherita, Italy, June

Zeigler B.P., Ball G., Cho H.J., Lee J.S. (1999) "Implementation of the DEVS Formalism over the HLA/RTI: Problems and Solutions", *Proc. Of SIW* Orlando, June

SUPPLY CHAIN VULNERABILITY AND RESILIENCE: A STATE OF THE ART OVERVIEW

Francesco Longo[(a)], Tuncer Ören[(b)]

[(a)]M&SNet (McLeod Modeling & Simulation Network)
Modeling & Simulation Center Laboratory of Enterprise Solutions (MSC-LES)
University of Calabria, Italy

[(b)]M&SNet (McLeod Modeling & Simulation Network)
SITE, University of Ottawa, Ottawa, ON, Canada

[(a)]f.longo@unical.it , [(b)]oren@site.ottawa.ca

ABSTRACT

The main objective of this paper is to provide the reader with a state of the art on the supply chain resilience. Markets globalization and global supply chains have to be regarded as business opportunities of economic development for each supply chain actor, but at the same time, they introduce a number of risks and vulnerabilities that affect the capability of the supply chains to maintain equilibrium states over long period of time. The importance of supply chain resilience and its connection with risks and vulnerabilities is underlined. Several case studies are presented in the context of supply chain resilience. Two different frameworks for categorizing supply chain risks are presented and the importance of information sharing and visibility along the supply chain is highlighted. Finally in the context of supply change management, Modelling & Simulation is presented as an ideal framework for experiencing critical events, understanding the effects of risks on supply chain vulnerability and testing supply chain resilience.

Keywords: Vulnerability, Security, Resilience, Supply Chain Management, Supply Chain Change Management

1. INTRODUCTION

Starting from 1990 a continuous markets globalization process has taken place involving supply chains and causing a remarkable extension of goods, services, information and financial flows. Before September 11 supply chain change management took into consideration above all robustness and costs efficiency neglecting or slightly considering supply chain risks, security and vulnerability. The supply chain robustness is defined as the capability to keep under control outputs variability (supply chain performances as service level, total costs and so on) in correspondence of high inputs variation (Christopher and Rutherford, 2004). Markets globalization, September 11 (or similar events), natural disasters effects (i.e. Katrina hurricane in New Orleans) have strongly underlined the need to consider, as further priority aspect in supply chain management, the "continuity" in correspondence of catastrophic events or disruptions. Supply chain disruptions happens under the effects of specific risks that can be internal to the company being considered, external to the company or external to the supply chain.

As reported in “Understanding Supply Chain Risk: A Self Assessment Workbook” (2003), terrorist attacks, wars, politic problems, natural disasters should be considered as risks external to the supply chain. Risks coming from the market or from suppliers should be considered as external to the company and internal to the supply chain; finally risks related to processes and activities should be considered as internal to the company (Sheffi, 2006; Sheffi, 2005-a).

In such a context the supply chain capability to assure continuity can be expressed in terms of resilience. The resilience is defined as the system ability to reach its equilibrium state (or another more desirable equilibrium state) after being disturbed by external or internal factors. As reported in “Creating a Resilient Supply Chains: A Practical Guide” (2003) the resilience definition takes into consideration the following aspects: supply chain flexibility, agility, velocity, visibility and redundancy. Before getting into details of the elements affecting the supply chain resilience, let us give a brief overview of the paper. Section 2 presents a list of research works focused on supply chain vulnerability, security and proposes some real case studies in the context of supply chain resilience. Section 3 describes the most important frameworks proposed in the literature for categorizing supply chain risks and highlights the importance of information sharing and visibility along the supply chain. Section 4 proposes the Modelling & Simulation based approach as decision support tool for improving supply chain resilience. Finally the last section reports the conclusions and the research activities still on going.

2. SUPPLY CHAIN RESILIENCE

2.1. Some definitions of resilience

According to the on-line *American Heritage dictionary*, resilience is the (i) “ability to recover quickly from illness, change, or misfortune"; (ii) "the property of a material that enables it to resume its original shape or position after being bent, stretched, or compressed".

According to the *on-line Merriam-Webster* the resilience is the (i) "an ability to recover from or adjust easily to misfortune or change"; (ii) "capability of a strained body to recover its size and shape after deformation caused especially by compressive stress".

The *on-line Compact Oxford Dictionary* defines resilience as the (i) "ability to withstand or recover quickly from difficult conditions (of a person)"; (ii) "the ability to recoil or spring back into shape after bending, stretching, or being compressed".

The definitions reported above refer to resilience as a property (of a person or material). Analogously we can say the resilience is a critical property that, in a context of supply chain change management, allows to the supply chain to react to internal/external risks and vulnerabilities quickly recovering an equilibrium state capable of guarantying high performance and efficiency levels.

2.2. Main factors affecting supply chain resilience

As reported into the introduction the most important elements affecting supply chain resilience are: flexibility, agility, velocity, visibility and redundancy.

Sheffi (2006) says that flexibility helps companies in correctly answering to markets variability and it can be obtained simultaneously using some factors as concurrent processes, final goods completing postponement inside the supply chain, strategies sharing with suppliers. The author also reports the case of the Hewlett-Packard printers: the completion of each printer (in terms of power supply, wall plug and instructions language) is delayed until the reception of the orders coming from different European countries. In that case the HP printer was ad-hoc designed for the postponement of the operations regarding power supply, wall plug and instructions language.

Christopher and Rutherford (2004), the report "Creating a Resilient Supply Chains: A Practical Guide" (2006) propose an accurate definition and description of the supply chain agility, velocity, visibility and redundancy. The authors define the agility as the company capability to quickly respond to unforeseen and unpredictable demand/supply markets changes. Note that the agility of a company also depends on the agility of all the actors involved in the supply chain. The velocity must be interpreted as time required for moving goods along the supply chain. The velocity is usually measured in terms of lead times. The visibility is the capability of the company to see all the information regarding the flow of products, information and finances both downstream and upstream along the supply chain. The redundancy is the augmentation of capacity and inventory in each node of the supply chain for facing supply chain disruption events. Note that, as underlined by Sheffi (2006), a company that hold extra inventory and capacity can incur in costs augmentation moving against all the principles dictated by just-in time approaches, lean processes and six-sigma methodology. Further aspects to be considered for supply chain resilience are the culture corporate and information sharing among the supply chain actors (Stenger et al., 2000).

2.3. Some research works on supply chain vulnerability

After the 9/11 attacks to the twin towers, the number of research works in the filed of supply chain security and vulnerability has strongly increased. In effect the 9/11 event is considered as one of the most important supply chain disruption event due to both short and long period economic effects.

In the sequel, we enumerate and provide short descriptions of some significant research works considering different types of risks affecting the supply chain. Even before 2001, the supply chain vulnerability and security were considered as an important topic as testified by the research work carried out by Choi and Hartley (1996), Christopher (1998), Braithwaite and Hall (1999). The first work presents and compares different methodologies for suppliers base selection; the second research work focalizes on strategic approaches for reducing costs within the supply chain; finally the last research work faces the problem of critical decisions (in terms of business risks) in supply chain management.

Peck and Juttner 2002 propose a survey on the main supply chain risks also considering the companies reactions for reducing/eliminating risks. Pai et al. (2003) present some methods for risks analysis based on Bayesian Networks, Fuzzy Logic and a combined approach of both.

A number of research studies introduce the concept of supply chain resilience. Such studies propose a map of supply chain vulnerabilities and classify supply chain risks in different phenomena (i.e. Hurricanes, Earthquakes), incidents (i.e. Exxon Valdese, Chernobyl), terrorist attacks (9/11 USA, 3/11 Spain), market globalization (i.e. strikes, new security procedures, insolvency).

For further information refer to Sheffi (2005-a), Sheffi (2005-b), Sheffi and Rice (2005), Christopher and Rutherford (2005), Sheffi (2006).

Concerning risk analysis and categorization, Wu et al. (2006) propose a risk analysis in the inbound supply chain identifying, evaluating and validating supply chain risks. Finally Gaonkar and Viswanadham (2004) present a model of a conceptual framework for supply chain risk categorization at strategic level.

Longo and Mirabelli (2008) focalize on the effects of demand/supply variability by using a supply chain management tool based on Modelling & Simulation (the aim is to understand the impact of such factors on each supply chain stage, i.e. distribution centers, stores, etc.). Still on demand/supply variability De Sensi et al. (2007) present and compare different inventory control policies considering market demand and lead times constraints. Nagurney and Matsypura (2005) propose a model of a three stages supply chain able to monitor network nodes competitiveness.

The scientific researches carried out by Sun and Yu (2005) and Deleris and Elkins (2004) respectively regard the impact of catastrophic events on supply chain contracts and probability distribution of losses caused by such events.

Further studies on supply chain vulnerability reduction regard the information sharing. In particular Sheffi (2005-a) and Suo and Jin (2004) states that one of the critical problems of the information sharing is the Bullwhip effect that is the amplification of the demand uncertainty moving back along the supply chain. Concerning the Bullwhip effect further information can be found in Lee et al. (1997).

It's important to stress that several scientific researches on supply chain vulnerability and risks management have involved big companies operating in the most important sectors or have been developed considering specific supply chain sectors. Hopper and Beck (2004) propose a study on supply chain risks in the automotive sector, Agrell et al. (2004) in the telecommunication sector. Some examples of research studies in specific supply chain sector regard marine security levels analysis in marine terminals (Barnes and Oloruntoba, 2005; Longo and Bruzzone, 2005; Longo et al. 2005, Bruzzone et al. 2005), costs evaluation for disruptive events in railways networks (Mirabelli et al. 2005), innovative tools for risk management in the aeronautical sector (Haywood and Peck, 2003) and the introduction of strategies for minimizing and contrasting the effects of the new products from foreign markets/countries in the textile sector (Chandra, 1999).

2.4. Supply Chains disruption: some real case studies

As mentioned in the previous section supply chain risks have to be regarded as critical factors affecting the supply chain management (Sheffi, 2005-a; Peck and Juttner 2002; Pai and Zhou 2003; Wu et al. 2006). Supply chain risks are situations of potential danger that could happen out of control affecting consistently supply chains, above all in case of just in time synchronized production sites in different parts of the world, stock minimization and lead time reduction. Sheffi (2006) reports clear examples of supply chains disruption. After the 9/11 terrorist attack, the U.S. government shutdown the Canadian and Mexican borders forcing Chrysler and Ford to an intermittent production. Toyota supply chain thanks to redundancy in terms of multiple suppliers and multiple stocks was more resilient than Chrysler and Ford supply chains. Toyota was capable of assuring a greater "continuity" in correspondence of the terrorist event and, above all, in correspondence of the U.S. government overreaction. An analogous situation was that after the Taiwan earthquake that stopped for a while the supply of semiconductors from that country. Also in this case the supply chains of some companies, such as Dell, were more reactive (more resilient) than other companies such as Apple.

Other examples of external risks causing supply chains disruption are strikes, suppliers' insolvency or fast withdrawal of already distributed products. Examples of fast withdrawal of already distributed products are: (1) the Mad Cow Disease (1996); (2) the high levels of Dioxin in Coca-cola drinks, Belgium (May 1997); (3) the high levels of Dioxin in Belgium Poultry (July 1999); (4) the diethylene glycol in the Colgate toothpaste (July 2007); (5) the Mattel Lead Contaminated toys (August 2007).

Consider as further examples of the impact of risks on supply chains vulnerability and resilience the Nokia-Ericsson case and the Land Rover case (further information can be found in "Creating a Resilient Supply Chains: A Practical Guide" (2003). The 17th of March 2000, due to a problem to the power lines of a Philips Electronics plant, millions of silicon wafers and chips for Nokia and Ericsson mobile phones were destroyed (at that time Philips was the sole supplier for both Nokia and Ericsson and the worldwide demand for mobile phones was booming). Philips communicated to Nokia and Ericsson that the problem was totally under control. Nokia decided to investigate deeply the problem and discovered that the situation was so critical that the supplies would be disrupted for months. Consequently Nokia started to ask additional supplies to the other Philips plants and decided to modify its mobile phones in order to include different types of silicon wafers and chips. On the contrary Ericsson did not investigate the problem, trusting its first and sole supplier. When Ericsson understood that the supplies were destroyed for months was too late for finding additional suppliers. The economic impact for Ericsson in terms of loss revenues were estimated in about 400 millions of dollars and Nokia cemented its position as leader of the mobile phones market.

In 2001 the sole supplier of Land-Rover for its Land Rover Discovery was the UPF-Thompson that, at the end of 2001 became insolvent. Land Rover was able to afford production stop but the economic impact was sudden and severe.

In both cases the main problem was the suppliers base strategy. Just-in time approaches and costs minimization require a low number of suppliers (the best situation is one supplier for each specific component or service). In this case the entire supply risk falls upon that supplier strongly affecting the supply chain resilience.

3. SUPPLY CHAIN RISK ANALYSIS

Reducing supply chain vulnerability and improving supply chain resilience requires to categorize and analyze risks as well as requires to understand the effect of information sharing on visibility along the supply chain.

3.1. Categorizing the supply chain risks

Mason et al. (1998) and Cristopher and Peck (2004) propose a framework for categorizing supply chain risks subdivided in five different categories:

- process risks internal to the company;
- control risks internal to the company;
- demand risks external to the company and internal to the supply chain;
- supply risks external to the company and internal to the supply chain;
- environmental risks external to the supply chain.

Note that a company is the union of different processes and activities that aim, in the long period, at increasing the value added of the business strategies. Process risks may affect all the activities carried out by the company, from the manufacturing production to quality levels, from warehouses management to transportation activities.

Control risks are strictly related to Process risks. In effect processes and activities are governed by rules and controls. The warehouse management is performed by using inventory control policies, the manufacturing process in a job shop is ruled by shop orders scheduling, the quality levels depends (among the others) on the methodology being used for improving quality. In other terms each process inside a company has specific controls and rules. Wrong controls and rules act as risks affecting the performances of the company and its resilience (i.e. wrong inventory control policies and/or demand forecast methodologies, inadequate production planning, wrong corporate culture during the implementation of quality methodologies and systems, etc.).

Demand risks usually involve the flow of products, information and finances downstream the company being considered. Such risks are related to the powerlessness of the company (due to unpredictable events) to satisfy market demand and also include demand forecasts risks and Bullwhip effect. Note that among the consequences of markets globalization the most important affecting the demand forecasts risks are the growing products assortment and the shorter products life cycle. In such a context classical demand forecasting techniques may result inadequate. In effect numerous research works have been proposed in order to consider higher items aggregation levels and more reliable forecast models (two different examples are respectively reported in Dekker et al., 2004 and Zotteri et al., 2005).

Supply risks involve the flow of products, information and finances upstream the company being considered. Such risks are related to suppliers' reliability and suppliers' base selection. Note that suppliers should be able to delivery the right products at the right place and time. The Nokia-Ericsson and the Land Rover cases (presented in section 2.1) show that the supply risks cause the supply chain disruption in case of supplier's insolvency or inability to deliver the materials over a long period of time.

Finally the environmental risks have to be regarded as uncontrollable and sometime unpredictable events that strongly affect the supply chain vulnerability and resilience. Among the others the most important are natural disasters, wars, terrorist attacks, political and social disorders. The 9/11 attacks in USA demonstrated the vulnerability of the U.S. economy to shutdown the transportation system, and especially the vulnerability of extended supply chains and trans-border just-in-time manufacturing systems. Consider for instance the case of container terminals security. Container terminals are the most important rings of the cargo supply chain. Before 9/11, about 2% of incoming containers were physically opened and inspected and this percentage has been increased to 5.4% with dramatic effects on supply chain performances.

Another alternative framework for categorizing supply chain risks can be found in "Creating a Resilient Supply Chains: A Practical Guide", (2003). The authors recognize four levels of risks, named as follows:

- Process and Value stream (first level);
- Assets and Infrastructure Dependencies (second level);
- Organizations and Inter-Organizational Networks (third level);
- Environment (fourth level).

The risks of the first level regard all the processes and the value added both upstream and downstream the company being considered. In effect in the first level the problem of the supply chain vulnerability and supply chain risks should be faced by considering the entire supply chain. For a better understanding of the first level risks consider that the "process risks", described in the framework proposed by Mason et al. (1998) and Cristopher and Peck (2004), have to be regarded as extended to the entire supply chain, applied to each actor of the supply chain. The reduction of the first level risks requires high levels of trust among the supply chain actors and in turn this means information sharing and high visibility along the entire supply chain.

The risks of the second level regard the assets and the infrastructure dependences. A supply chain is made up by links and nodes. In terms of products flows, nodes are distribution centers, plants, terminals, stores, whilst links are roads, waterways, rails, etc. In terms of information flows, nodes are ICT platforms while links are the communication networks that connect, at each level (national, international, intercontinental), the ICT platforms. The continuity of the operations in each node and/or link (risks reduction and resilience enhancement) should be assured by all the managers, operators and workers at each stage of the supply chain. The second level risks underline the importance of the human factor for supply chain management.

The risks of the third level regard the supply chain strategic management. The relationships between the supply chain actors are ruled by the position of power of each actor. In a globalized market the high levels of competitiveness usually lead companies to fight each other even in the same supply chain, pursuing different objectives or abusing of the own position of power. Once again the case Nokia-Ericsson shows how Philips (the sole supplier of both Nokia and Ericsson abused of its position by minimizing the problem to its plant).

Finally the risks of the fourth level are the same environmental risks described in the framework proposed by Mason et al. (1998) and Cristopher and Peck (2004).

A toolkit for supply chain risks categorization, analysis can be found in "Creating a Resilient Supply Chains: A Practical Guide" (2003). Among the others, the authors propose Scenario Planning, Brainstorming, Failure Mode and Effects Analysis, Flowcharting, Pareto Analysis, Modelling & Simulation as powerful tools that can help the managers in supply chain risks management.

4. SUPPLY CHAIN CHANGE MANAGEMENT BASED ON MODELING & SIMULATION

4.1. The Supply Chain Change Management

It is now well clear that supply chain is continuously changing over the time. A part of the effort of supply chain managers should be devoted to supply chain change management. Each new business opportunity is always characterized by risks that strongly affect supply chain vulnerability and resilience. In effect the supply chain change management should pursue optimal trade-offs between technical-economic advantages (costs reduction, productivity increase) and resilience (considered as risk levels and vulnerability variation) by considering the company and its processes, the rules and the controls, the organizations, the infrastructures, the business strategies and the environmental conditions. Such optimal trade-offs can be obtained by performing the following actions:

- tactical, strategic and operative analysis of the decisions tools used for supply chain change management (from the perspective of resilience);
- enhancement of the decisions tools for supply change management including specific toolkits for risks categorization and analysis;
- development of decisions models and operative tools for supply chain change management in order to reduce vulnerability and increase resilience and security;
- scenarios planning and development for decisional models and operative tools verification, validation and testing (VV&T).

Taking into consideration such actions, the supply chain change management devoted to support and improve supply chain resilience should be subdivided in four main stages. The first step is a survey on supply chain change management considering strategic business decisions and the effects of such decisions on supply chain vulnerability. The second stage identifies the actual guidelines followed in supply chain vulnerability management. According to these phases it will be possible to categorize risks at every level as well as to guarantee, using crossing and comparative analyses, existing tools improvement or to introduce new tools in order to increase supply chain flexibility and agility.

The third stage proposes the integration of the previous results together with methodologies capable of guarantying supply chain low risks and vulnerability levels in order to develop decisional models and operative tools for supply chain change management.

To better understand the fourth and last step, it is necessary to underline that a change process pushes the company to migrate from the actual scenario or context to a different one; the characteristics of such new scenario depend on controllable and uncontrollable factors (for example, the market evolution dynamics is an uncontrollable factor while manufacturing process reliability is a controllable factor). As consequence, the definition and the analysis of the change processes must develop several scenarios and/or evolutionary contexts that have to be used as case studies for verification, validation and testing of the decisional models developed.

4.2. Modeling & Simulation for supporting supply chain resilience

Modeling & Simulation (M&S) provides an ideal framework for experiencing critical events, understanding the effects of risks on supply chain vulnerability, testing scenarios and evolutionary contexts, measuring supply chain resilience.

The development of a decision support tool, based on M&S to be used for supporting the enhancement of supply chain resilience should follow the four steps described in the previous section. The change processes management should take into consideration the identification of cause-effect relations that connect strategic business choices to elements of vulnerability, security and resilience of supply chains. After the development of decision models for supply chain change management in the framework of resilience a framework capable of hosting the decision models should be implemented, opportunely integrated in modelling and simulation tools. A simulator should combine the different models to operate as a complete and process integrated decisional tool (i.e. a federation in an High Level Architecture-HLA environment capable of integrating the modules such as demand forecasting, logistic flows, production risks, etc.).

Longo & Massei (2008) propose the architecture of an M&S framework based on HLA for integrating different decision models for supporting supply chain resilience and vulnerability. Bruzzone et al. (2006) propose a demonstrator that uses M&S for providing and analyzing a crisis scenario of hurricane event. The simulation allows to understand the relations between the crisis scenario and the transportation activities with special attention to logistic flows and indirect costs.

5. CONCLUSIONS

The paper proposes an exhaustive state of the art overview on supply chain vulnerability and resilience. The study of the research works developed during the last years, the analysis of the real case studies allows to understand that supply chain resilience is a quite complex topic involving different research area. Among the others the most important research areas are:

1. Supply chain vulnerability, security and resilience management
2. Methods for demand forecasting and supply risks analysis in supply chain
3. Information management and visibility along the supply chains
4. Supply chain Life Cycle Costing
5. Modelling & Simulation devoted to support supply chain resilience.

In addition note that the concept of resilience becomes more important for Small & Medium Enterprises (SMEs). In effect big companies usually have management tools for facing supply chain risks and reducing vulnerabilities. The most challenging objective is to develop an integrated tool that allows SMEs to react in an agile, lean and flexible way to the events that characterize the evolution of competitive and international markets (both internal or external to the supply chain, controllable or not, unexpected, destructive or catastrophic).

REFERENCES

Agrell, P. J., Lindroth, R., Norrman, A., 2004. Risk, information and incentives in telecom supply chains, *International Journal of Production Economics* 90, pp. 1-16.

American Heritage Dictionary: resilience, Available from: http://www.merriamwebster.com/dictionary/resilience

Barnes, P., Oloruntoba, R., 2005. Assurance of security in maritime supply chains: Conceptual issues of vulnerability and crisis management. *Journal of International Management*, 11 pp. 519-540.

Braithwaite, A., Hall, D., (1999). Risky business? Critical Decisions in supply chain management, Part 2, *Supply Chain Practice*, 1(3), pp. 44-58.

Braithwaite, A., Hall, D., 1999. Risky business? Critical Decisions in supply chain management, Part 1, *Supply Chain Practice*, 1(2), pp. 40-47.

Bruzzone, A. G., Longo, F., Papoff, E., 2005. Metrics for global logistics and transportation facility information assurance, security, and overall protection. *Proceedings of the European Simulation Symposium*.

Bruzzone, A.G., Longo, F., Massei, M., Saetta, S., 2006. The vulnerability of supply chains as a key factor in supply chain management. *Proceedings of the Summer Computer Simulation Conference*, Calgary (Canada).

Chandra, P., 1999. Competing through capabilities: strategies for global competitiveness of the Indian textile industry. *Economic & Political weekly*, 23.

Choi, T. Y., Hartley, J. L., 1996. An exploration of supplier selection practices across the supply chain. *Journal of operations Management*, 14 pp. 333-343.

Christopher, M., 1998. Logistics and Supply Chain Management: Strategies for reducing costs and Improving Service, *Financial Times, Prentice Hall*.

Christopher, M., Rutherford, C., 2004. Creating a supply chain Resilience through Agile Six Sigma, *Critical Eye Publications LTD*, pp. 24-28.

Christopher, M.., Peck, H., 2004. Building the resilient supply chain. *International Journal of Logistics Management*, 15(2), pp. 1-13.

Compact Oxford Dictionary: resilience. Available from: http://www.askoxford.com/dictionaries/compact_oed/?view=uk

Creating a Resilient Supply Chains: A Practical Guide, 2003. Report produced by the *Centre for Logistics and Supply Chain Management, Cranfield school of Management*.

De Sensi, G., Longo, F., Mirabelli, G., 2007. Inventory policies analysis under demand patterns and lead times constraints in a real supply chain. *International Journal of Production Research*, doi: 10.1080/00207540701528776.

Dekker, M., Van Donselaar, K., Ouwehand, P., 2004. How to use aggregation and combined forecasting to improve seasonal demand forecasts. International Journal of Production Economics, 90 pp. 151-167.

Deleris, L. A., Elkins, D., 2004. Analyzing Losses From Hazard Exposure: A Conservative Probabilistic Estimate Using Supply Chain Risk Simulation, *Proceedings of the Winter Simulation Conference*.

Gaonkar, R., Viswanadham, N., 2004. A Conceptual and Analytical Framework for the Management of Risk in Supply Chains. *Proceedings of the IEEE lnternational Conference on Robotics & Automation*, New Orleans.

Haywood, M., Peck, H., 2003. Improving the management of supply chain vulnerability in UK aerospace manufacturing. *Proceedings of the 1st Euroma/POMs Conference*, pp.121 – 130.

Hopper, S., Beck, M., 2004. Supply Chain Risk: A Global Automotive Industry Viewpoint. *International Corporate Rescue*, 1(4), pp. 23-26.

Lee, H. L., Padmanabhan, V., Whang, S., 1997. The Bullwhip Effect in Supply Chains. *Sloan Management Review*, 38(3), pp. 93-102.

Longo, F., Bruzzone, A. G., 2005. Modeling & Simulation applied to Security Systems. *Proceedings of the Summer Computer Simulation Conference*, pp. 183-188.

Longo, F., Massei, M. 2008. Advanced Supply chain Protection & Integrated Decision support System. *Proceedings of the Asian Modeling Symposium,* Kuala Lumpur (Malesya).

Longo, F., Mirabelli, G., 2008. An Advanced Supply Chain Management Tool Based on Modeling & Simulation, *Computer and Industrial Engineering*, 54(3), pp 570-588.

Longo, F., Mirabelli, G., Viazzo, S., 2005. Simulation and Design of Experiment for analyzing security issues in container terminals. *Proceedings of the International Workshop on Modelling and Applied Simulation.*

Manson-Jones, R., Towill, D., 19989. Shrinking the supply chain uncertainty cycle, *Control*, pp. 17-22.

Merriam-Webster Dictionary: resilience. Available from: http://www.bartleby.com/61/

Mirabelli, G., Papoff, E., Viazzo, S., 2005. Conceptual Model for Analysis of Costs/Risks/Quality within Railway Activities. *Proceedings of the International Workshop on Harbour Maritime & Multimodal Logistics Modelling & Simulation.*

Nagurney, A., Matsypura, D., 2005. Global supply chain network dynamics with multicriteria decision-making under risk and uncertainty. *Transportation Research*, Part E, 41(6), pp. 585-612.

Pai, R. R., Kallepalli, V. R., Caudill, R. J., Zhou, M., 2003. Methods Toward Supply Chain Risk Analysis Systems. *Proceedings of the International Conference on Man and Cybernetics*, 5, pp4560 – 4565.

Peck, H., Juttner, U., 2002. Risk Management in the supply chain. *Focus*, pp. 18-200.

Sheffi, Y., 2005-a. Building a Resilient Supply Chain *Harvard Business Review*, 1(8), pp. 1-4.

Sheffi, Y., Preparing for the Big One, 2005-b, *IEE Manufacturing Engineer.*

Sheffi, Y., 2006. Resilience reduces risk. *The Official Magazine of The Logistics Institute*, 12(1), pp. 13-14.

Sheffi, Y., Rice, J., 2005. A supply chain view of the Resilient Enterprise, *MITSloan Management Review*, 47(1), pp. 41-48.

Stenger, A. J., Ganeshan, R., Boone, T., 2000. The integration aspect of Supply Chain Management: a framework and a simulation. *Supply Chain Management: innovations for education, POMS Series in Technology and Operations Management* 2, pp. 141-156.

Sun, C., Yu, H., 2005. Supply Chain Contract under Product Cost Disruption. *Proceedings of the International Conference on Services Systems and Services Management*, 1, pp. 708 – 711.

Suo, H., Jin, Y., 2004. Supplier-Retailer Contracting Under Asymmetric Risk Attitude Information in Supply Chain. *Proceedings of the 2004 IEEE International Conference On Networking, Sensing & Control.*

Understanding Supply Chain Risk: A Self Assessment Workbook, 2003. *LCP Consulting and Centre for Logistics and Supply Chain Management, Cranfield school of Management.*

Wu, T., Blackhurs, J., Chidambaram, V., 2006. "A model for inbound supply risk analysis", (2006) Computers in Industry.

Zotteri, G., Kalchschmidt, M., Caniato, F., 2005. The impact of aggregation level on forecasting performance. International Journal of Production Economics, 93-94 pp. 479-491.

AUTHORS BIOGRAPHIES

FRANCESCO LONGO took the degree in Mechanical Engineering from University of Calabria (2002) and the PhD in Industrial Engineering (2005). He is currently researcher at the Mechanical Department (Industrial Engineering Section) of University of Calabria. His research interests regard modeling & simulation of manufacturing systems and supply chain management, vulnerability and resilience, DOE, ANOVA. He is Responsible of the Modeling & Simulation Center – Laboratory of Enterprise Solutions (MSC-LES), member organization of the MS&Net (McLeod Modeling & Simulation Network) He is also member of the Society for Computer Simulation International and Liophant Simulation.

TUNCER ÖREN is a professor emeritus of Computer Science at the University of Ottawa. His current research activities include (1) advanced M&S methodologies such as: multimodels (to encapsulate several aspects of models), multisimulation (to allow simultaneous simulation of several aspects of systems), and emergence; (2) agent-directed simulation; (3) cognitive simulation (including simulation of human behavior by fuzzy agents, agents with dynamic personality and emotions, agents with perception, anticipation, and understanding abilities); and (4) reliability and quality assurance in M&S and user/system interfaces. He has also contributed in Ethics in simulation as the lead author of the Code of Professional Ethics for Simulationists, M&S Body of Knowledge, and multilingual M&S dictionaries. He is the founding director of the M&SNet of SCS. He has over 350 publications (some translated in Chinese, German and Turkish) and has been active in over 370 conferences and seminars held in 30 countries. He received "Information Age Award" from the Turkish Ministry of Culture (1991), Distinguished Service Award from SCS (2006) and plaques and certificates of appreciation from organizations including ACM, AECL, AFCEA, and NATO; and is recognized by IBM Canada as a Pioneer of Computing in Canada (2005). His home page is: http://www.site.uottawa.ca/~oren/.

INCORPORATING "BIG FIVE" PERSONALITY FACTORS INTO CROWD SIMULATION

Sivakumar Jaganathan, J. Peter Kincaid, Thomas L. Clarke

Institute for Simulation and Training
University of Central Florida, 4300 Technology Parkway, Orlando, FL 32826

shiva@mail.ucf.edu, pkincaid@ist.ucf.edu, tclarke@ist.ucf.edu

ABSTRACT

The simulation and modeling of crowd behavior has become an active research area in recent years. This area of research has been applied to a wide variety of domains such as military, education, training, entertainment and human factors analysis. Most crowd simulations do not consider the effects of cultural or personality diversity within the crowd. We incorporate these effects by modifying the social force terms within the Helbing-Molnar-Farjas-Vicsek (HMFV) crowd model implemented within the MASON (Multi-Agent Simulation of Neighborhoods) environment. The modification are based on the "Big Five" personality factors (neuroticism, extroversion, openness, agreeableness and conscientiousness) which have been found to be applicable across cultures.

In addition to detailing the modifications, this paper reports on comparison of the Big Five modifications of the HMFV model to videos of crowds. An expert panel of behavioral scientists found the modified HMFV simulations to be realistic models. In addition a preliminary version of a technique used based on optical flow analysis of the videos showed good correlation.

Keywords: Crowd Simulation, HMFV, Big Five, Optical Flow, Personality interaction

1. BACKGROUND

Recently, a considerable amount of research has been performed on simulating the collective behavior of pedestrians in the street or of people finding their way inside a building or a room. A comprehensive review of the state of the art can be found in Batty et al (2002). In all these simulation studies, one area that has been lacking is accounting for the effects of human personalities and cultural characteristics on the outcome.

Existing crowd models such as Helbing-Molnár-Farkas-Vicsek (HMFV) [Helbing et al 2002] do not take individual personality and cultural factors into account in attempting to model and simulate human crowds, Nevertheless, it is recognized that personality, in addition to other factors, is a significant determinant of individual behavior [Helbing and Molnar, 1995].

2. ADDITION OF PERSONALITY AND CULTURAL EFFECTS

The most popular approach among psychologists for studying personality traits is the Big Five factors or dimensions of personality [Soldz et al 1999 and Saucier et al 1998]. The five factors are neuroticism, extroversion, openness, agreeableness and conscientiousness.

A quantitative model has been developed based on the Big Five [Jaganathan, 2007]. An individual's personality composition is represented as a column vector. In addition, a personality interaction matrix (MPF) is also implemented which defines the interaction parameter between different types of personalities. The interaction between two individuals, Ind1 and Ind2 with their personality combinations, is modeled as a quadratic form:

$$interaction=[Ind1.vpers]^{transpose}[MPF]\{Ind2.vpers]$$

Where MPF is a 5 by 5 matrix describing the interaction between different personality factors.

To implement the simulation, the MASON multi-agent simulation toolkit was used. This framework is available for developers from the George Mason University (cs.gmu.edu/~eclab/projects/mason/). The interpersonal social forces used in the HMFV model were modified with a personality dependent multiplier as outlined above.

3. COMPARISON WITH CROWD OBSERVATIONS

A face validity measure was used as reported in Jaganathan [2007]. Face validity is concerned with how a measure or procedure appears generally to a group of subject matter experts. In this research the establishing of face validity, provided a reasonable way to gain confidence in the research methods and the tools that were developed as part of this research.

The expert panel consisted of three Ph.D. behavioral scientists all of whom have significant experience in research relating to behavior and personality. The research experience of the three panelists ranges between 10 and 40 years. The panelists were interviewed individually, but the analysis below is a combined summery of the points that they made. The panelists commented on the videos and simulations as they were shown and then made some additional observations. They remarked favorably on how the personality modified HMFV model could capture aspects of the crowds leaving the church.

The optical flow analysis of the crowd videos similar to that reported in Clarke et al [2007] also agreed with the results of the simulation confirming both the validity of the Big Five HMFV model and the observations of the expert panel.

4. CONCLUSION

By adding personality factors to the base simulation we have developed a significant modification to the HMFV model. This incorporation of Big Five personality factors into the base HMFV model has helped make the individuals in the crowd behave more realistically. Grouping and sub grouping of individuals within the crowds demonstrate that there are some social skills exhibited by the individuals in the crowd. For example, the simulation depicting a crowd exiting a church, was considered to be a highly social situation in which individuals would show high levels of the personality factor of extroversion.

Ongoing research aims at considering individual differences in age, gender, culture and ethnicity to increase the value of crowd simulation both in operational and training contexts. The explicit incorporation of gender, ethnicity, age, and cultural differences as factors in the model will ensure broad applicability of the research to crowd control in nations other than the U.S. Cultural differences have a significant effect on personal space and thus the interpersonal forces.

ACKNOWLEDGEMENTS

We thank B. Goldiez R. Shumaker, and Jim Szalma of the Institute for Simulation and Training for their interest and discussions throughout different stages of this work. We also thank F. Jentsch for help with translating Weidmann (1992).

This research has been supported in part by the National Science Foundation under Grant No. BCS-0527545 and by the U.S. Army Research Development and Engineering Command, Simulation Technology Center, Contract N61339-02-C-0107.

REFERENCES

Batty, M., J. DeSyllas, and E. Duxbury. 2002. The discrete dynamics of small-scale spatial events: Agent-based models of mobility in carnivals and street parades, http://www.casa.ucl.ac.uk/workingpapers/Paper56.pdf.

Clarke, Thomas L. , D. J. Kaup, Linda Malone, Rex Oleson, Mario Rosa, 2007, Crowd Model Verification Using Video Data, EMSS 2007.

Helbing, D., I. J. Farkas, P. Molnar, and T. Vicsek. 2002. Simulation of pedestrian crowds in normal and evacuation situations, in *Pedestrian and Evacuation Dynamics*, edited by M. Schreckenberg and S. Deo Sarma, pp. 21-58, Springer-Verlag, Berlin.

Helbing, D. and P. Molnar,. 1995. Social force model for pedestrian dynamics, Phys.Rev. E, 1995, 51, pp. 4282-4287.

Intel. 2001. Open Source Computer Vision Library: Reference Manual. Intel, Santa Clara, CA.

Jaganathan , Sivakumar, 2007, On the Incorporation of the Personality Factors Into Crowd Simulation, UCF Phd Dissertation.

Kaup, D. J., J. E. Fauth, L.J. Walters, L. Malone, T. Clarke. 2006. Simulations as a mathematical tool, SIAM Conference on the Life Sciences, Raleigh, NC, Aug. 3, 2006.

Saucier, G., & Goldberg, L. R. (1998): What is beyond the Big Five? Journal of Personality, 66, 495-524.

Soldz, S., & Vaillant, G. E. (1999): The Big Five personality traits and the life course: A 45-year longitudinal study. Journal of Research in Personality, 33, 208-232.

BIOGRAPHIES

SIVAKUMAR JAGANATHAN is a post doc in Modeling and Simulation at the University of Central Florida with masters degrees in both Mechanical Engineering and Modeling and Simulation. He has worked at UCF both as a research engineer and as a data administrator. His current research involves modeling decision making processes in biological systems including crowd simulation.

J. PETER KINCAID is a Graduate Research Professor at the Institute for Simulation and Training and founding director of the UCF graduate program in Modeling and Simulation. He has worked as a faculty member at two universities, as a research scientist for the military, and as a human factors engineer for Martin-Marietta (now Lockheed Martin). His Ph.D. from the Ohio State University is in human factors psychology. .For the last 30 years, his research has focused on simulation-based training. Dr. Kincaid is a Certified Modeling and Simulation Professional (CMSP) charter member.

THOMAS L. CLARKE is Principal Mathematician at the Institute for Simulation and Training of the University of Central Florida and is also Associate Professor in the Modeling and Simulation program. He has a PhD in Applied Mathematics from the University of Miami and worked at NOAA before coming to UCF. He has a diverse background in applying mathematics to simulation and has investigated areas such as the application of catastrophe theory and nonlinear dynamics.

STUDY OF BIOLOGICALLY-INSPIRED NETWORK SYSTEMS: MAPPING COLONIES TO LARGE-SCALE NETWORKS

Ahmet Zengin[(a)], **Hessam Sarjoughian**[(b)], **Hüseyin Ekiz**[(c)]

[(a), (c)] Sakarya University Technical Education Faculty
Department of Computer Science Education
Serdivan, Sakarya, TURKEY
[(b)]Arizona State University
School of Computing & Informatics
Arizona Center for Integrative Modeling and Simulation
Tempe, AZ, USA

[(a)]azengin@sakarya.edu.tr , [(b)]sarjoughian@asu.edu, [(c)]ekiz@sakarya.edu.tr

ABSTRACT

Natural systems can offer important concepts for modeling network systems. A biologically-inspired discrete-event modeling approach is described for studying networks' scalability and performance traits. Key adaptive and emergent attributes of honeybees and their societal properties are incorporated into a set of simulation models that are developed using the Discrete Event System Specification approach. We describe our approach which is based on mapping the behavior of the honeybees to discrete event models. Large-scale network models are simulated and evaluated to show the benefits of nature-inspired network models.

Keywords: beehive, DEVS, networks, routing, scalability, simulation.

1. INTRODUCTION

Network systems must communicate with one another using a variety of algorithms and technologies. Many systems supporting interconnectivity are required to exhibit essential traits such as adaptability, scalability, and reliability (survivability). At the same time, these networked systems are expected to offer new and more sophisticated services in the face of increasing system heterogeneity (Lunceford and Page 2002). To cope with the management of such networks in the presence of ever increasing complexity, various decentralized and centralized approaches are being used to address the needs of private and public organizations (Steenstrup 1995).

Ecological models are being studied for development and operation of decentralized (distributed) systems such as communication networks. Large-scale biological systems, such as bee colonies, have advanced mechanisms that are scalable and adaptable under varying environmental conditions (Bonabeau, Dorigo, and Theraulaz 1999). The desirable characteristics of the bee colony, scalability, adaptability and survivability, are not present in any single bee. Rather, they emerge from the collective actions and interactions of all bees in the colony.

The design of complex and scalable network applications, therefore, stands to benefit from the power of biological principles and schemes. The focus of this paper is on applying biological principles and mechanisms to the design and implementation of network applications. Swarm-based routing algorithms offer a number of attractive features including autonomy, robustness and fault tolerance. Distributing intelligence on the network provides rapid control over resources that can dynamically adapt to user's requirements. Such swarm-based algorithms adapt well to dynamic topologies and compared to the current state-of-the-art distance vector algorithm have been shown to result in the highest throughput and lowest delays in Internet-style networks. We have devised a new class of agent-based routing algorithm based on principles of biological swarms, which have the potential to address some of the above problems in an autonomous and intelligent fashion.

To develop and study dynamic and adaptive swarm-based routing protocols, we have devised a DEVS (Discrete Event System Specification) (Zeigler, Praehofer, and Kim 2000) network model called SwarmNet. Discrete event simulation is a resilient, powerful and efficient computational device that support exploring the behaviors of complex systems. These models and networks are implemented in the DEVSJAVA modeling and simulation environment (Zeigler and Sarjoughian 2003, ACIMS 2008) which is an implementation of the DEVS framework.

The nodes and links are characterized as the elementary network components. Using the DEVS hierarchical model composition concept, we develop simulation models of networks with varying topologies and scales. For example, we will use clusters to study its impact on reducing communication and increasing performance. The explicit and hidden behaviors of these networks are observed under various experimental configurations – e.g., nodes and links are assigned different capacities. Lastly, we also examine the performance of the DEVSJAVA environment for networks having from tens to several thousands of components and connections.

In the remainder of this paper, starting in Section 2, we briefly review some related key features of SwarmNet and highlight its strengths and weaknesses. In Section 3, we describe the modeling concepts from honeybees are mapped to a set of adaptable agent-based DEVS modeling constructs. In Section 4, we develop and analyze example models in the SwarmNet simulation environment. In Section 5, we summarize our work and present some future research directions.

2. KEY FEATURES OF THE SWARMNET

Following treatment summarizes key features of the SwarmNet which is packet-level discrete event network simulator based on DEVSJAVA. It utilizes DEVS formalism for describing network components and inherits DEVS hierarchical and modular design concepts.

DEVS Simulation Engine: The dynamics of network systems can be described using discrete event modeling. This is because the dynamics of network systems can be characterized in terms of components that can process and generate events. Among discrete event modeling approaches, the Discrete Event Systems Specification (DEVS) (Zeigler, Praehofer, and Kim 2000) is well suited for formally describing concurrent processing and the event-driven nature of arbitrary configuration of nodes and links forming network systems. This modeling approach supports hierarchical modular model construction, distributed execution, and therefore characterizing complex, large-scale systems with atomic and coupled models. Atomic models represent the structure and behavior of individual components via inputs (X), outputs (Y), states (S), and functions (X, S, Y, δext, δint, δconf, λ, ta). The external (δext), internal (δint), confluent (δconf), output (λ), and time advance functions (ta) define a component's behavior over time (for examples of the atomic model see the Listings 1 and 2). Internal and external transition functions describe autonomous behavior and response to external stimuli, respectively. The time advance function represents the passage of time. The output function is used to generate outputs.

Atomic models can be coupled together in a strict hierarchy to form more complex models. Parallel DEVS, which extends the classical DEVS, is capable of processing multiple input events and concurrent occurrences of internal and external transition functions. The Parallel DEVS confluent transition function provides local control by handling simultaneous internal and external transition functions. A coupled model can be constructed by composing models into hierarchical tree structures. A coupled model is defined in terms of its constituent atomic and/or coupled models. Computational realizations of the DEVS formalism and its associated simulation protocols are executed using simulation engines such as DEVSJAVA (ACIMS 2008). DEVSJAVA is an object-oriented realization of Parallel DEVS. It supports describing complex structures and behaviors of network systems using object-oriented modeling techniques and advanced features of the Java programming language. The formal foundation of DEVS, its efficient execution, and the availability of sequential, parallel, or distributed simulation engines using alternative computational environments such as CORBA, HLA, and Web-services are important considerations. Furthermore, the DEVS models are extended with other kinds of models such as fuzzy logic (Sarjoughian and Cellier 2001).

Network Modeling Approach: DEVS makes modeling effort systematic so that complex behavior is formed by coupling simple structured primitive models (e.g. atomic model). In other words, behavior of an atomic model does not exhibit a high level intelligence; nevertheless coupled model shows fascinating emergent behavior. Atomic models in a compact model interact via messages to form complex collective behavior. In order for modeling a distributed networked system, we have defined a set of basic network simulation model components including nodes which communicate with one another via links. By coupling these model components in DEVSJAVA, we can develop a variety of network configurations and study network characteristics. Since it is assumed that only nodes and links of a network are able to cause bottleneck, they are modeled as atomic models and only their states as well as input and output variables are of interest. Other network components such as packets and routing tables are realized and modeled as stateless entities.

Modular and Hierarchical Design: A coupled model specifies constructs for composing modular models into hierarchical structures. Behavior of a coupled model is defined by its constituent atomic (and/or coupled) models. With closure under coupling feature of DEVS, coupled models can be used as atomic models in a larger model. Coupled models can be constructed systematically using the concepts of ports and couplings between them. When a component sends messages via its output ports, the couplings relay the messages to their designated input ports (Wymore 1993). Upon receipt of messages by atomic models, they immediately process these messages which may result in new states and generation of outputs.

Robust and accurate analysis and testing framework: To define simulation objectives, we utilize the concept of experimental frame an experimental frame to define the conditions under which a model can be experimented with and observed. Topologies of network models or Internet core networks can be evaluated in terms of their critical network components and their dependencies. In this work, a typical experimental frame consists of generator and transducer atomic models. We employ generator in order to create network traffic and to schedule special events such as unavailability of links or nodes. To realize this, a generator model sends messages to all the appropriate network components. Defined experimental frame is used for testing the model under various conditions and observing its behavior. Having been equipped with an experimental frame, simulation is run in under specific experimental conditions and results are observed. The results are then evaluated in terms of whether or not they are within in acceptable range; otherwise, the model parameters are changed and simulation experiments are repeated.

Bio-inspired robust, adaptive, scalable and collaborative design: As stated above, a swarm routing approach which is biologically inspired model is derived based on honeybees and their interactions called honeybee scout-recruit mechanism during foraging. In this framework, the movement of specialized packets such as artificial bees (called scouts) can be used to balance network loads. Given the similarity of this and agent-based approaches, each node is capable of accommodating an ensemble of scouts for controlling congestion in a distributed environment. In our implementation, analogous to honeybee scout-recruit system, each network node is considered as a beehive so that bees leave their hives for gathering nectar. Network corresponds to the world of honeybees who seek richer nectar sources, finding paths with higher capacities to result in nectar sources that have higher profitability. Control packets in the form of light-weight scout entities searching for nectar and foraging for information to aid survival of the network (honeybee colonies). Each hive deploys a number of scouts to find the most profitable paths for a given destination. Each router then uses the information received from all the nodes in the network obtained by its scouts to calculate the shortest path to each destination in terms of a chosen metric. Scouts control congestion by making alterations to routing tables in order to route new traffic away from congested nodes. Then, packets are dispatched from a source to a destination according to information gathered by scouts.

Cluster-based hierarchical routing: One of the main criteria for appreciating the network simulators is scalability. A network or simulator model is considered scalable with respect to network size, if simulation deserves its run properly while the number of network components such as nodes and links grows constantly. Because Internet should be designed in a hierarchical manner for to be better managed, hierarchy is needed for scalability (Zeigler and Mittal 2002). Cluster-based hierarchical routing was invented for making memory usage lesser of simulations over very large topologies. A network topology is composed of several layers in a hierarchical manner, thus shrinking routing table size. To be able to make use of hierarchical routing for the simulations, there is need for defining hierarchical topology and hierarchical addressing. In this study, we employed a clustering approach to support scalability and implemented when coupling the models. Clustering provides manageable network sizes by abstracting a subnet to single node in a higher level network. By considering a coupled model as an atomic model, DEVS coupled model concept has a resemblance with clustering. There exists a hierarchy of networks within the total of all nodes and routers. Each coupled model has a number of border nodes which are used for connecting it to other coupled networks. In our approach, clustering is done in addressing level of nodes. Hierarchical and modular structure of DEVS formalism facilitates implementation of clustering approach. Border nodes have an additional routing table consisting of the cluster names. This approach substantially decreases the information stored in routers.

3. SwarmNET FRAMEWORK

We have defined a set of basic network simulation model components including nodes which communicate with one another via links (see Figures 1) as detailed next. By coupling these model components in DEVSJAVA, we can develop a variety of network configurations and study their characteristics. Since it is assumed that only nodes and links of a network are able to cause bottleneck, they are modeled as atomic models and only their states as well as input and output variables are of interest. Other network components such as packets and routing tables are realized and modeled as stateless entities.

3.1. Network Model Specification

The nodes in the network are modeled as a DEVS atomic component. Each node has several inputs and outputs through which messages among nodes can be received and sent (see Figure 1). IP address as a unique id, unique name or code identifying each computer and user is assigned to every node in the network so that a packet can be directed to a specific destination. IP addresses also specify the location of a router in the network. The main part of the nodal structure is the network interface (NIC) that provides fundamental inter-networking services.

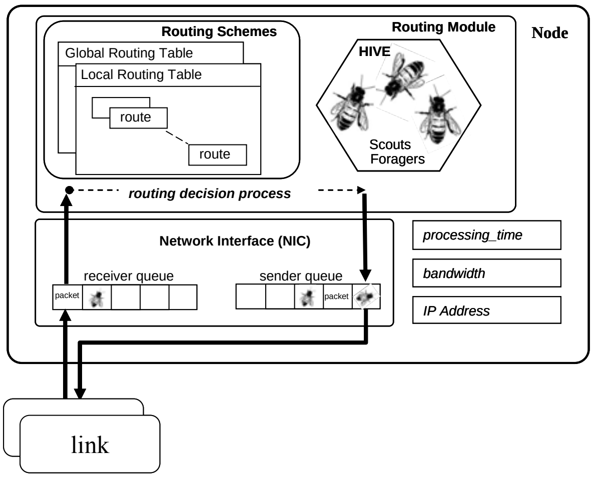


Figure 1: Node Model

At each node, packets are forwarded to their destination by using information stored in its Routing

Module, which defines a node’s routing capability and intelligence. Routing Module includes a routing table for local network as well as a global routing table which can be used to manage the routing between the local network and other parts of the global network. In our swarm application, beehive is configured to launch scouts, foragers, and other bees to monitor and reconfigure network resources. All link models are capable of accommodating different kind of entities for supporting different network designs.

The link is modeled as an atomic model. It has a central role in defining networks having different topologies. All links are communication channels and therefore are viewed as bit-pipes which are characterized with bandwidth (bits/sec) and transmission or propagation delay specified in milliseconds (see Figure 2). Each link is defined to be bidirectional and thus supports concurrent bidirectional interactions. Each duplex link has some finite capacity. The packets that arrive are placed in the queues and are transmitted to the next node using first-in first-out (FIFO) strategy. Links are able to carry traffic of a certain bandwidth up to the total capacity of the link. The specification of the link is akin to a simple processor with a queue that can process incoming packets according to FIFO or some other discipline. Each link has input and output ports for connecting two nodes in a duplex manner (see for example Link 1 atomic model in Figure 2).

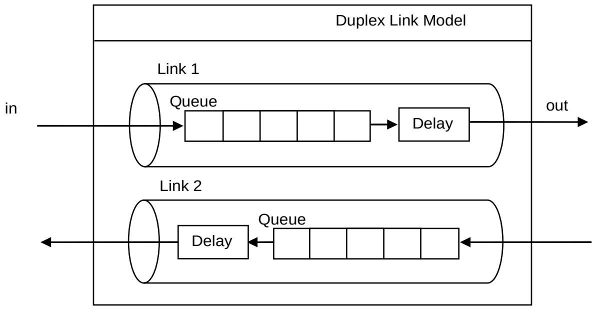
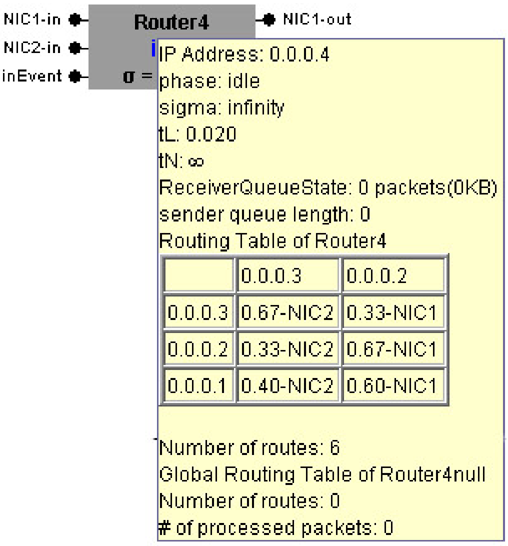


Figure 2: Bidirectional Link Model

Other network components such as routing tables and packets modeled in DEVSJAVA as stateless models. In Figure 3, the DEVSJAVA viewer shows the content of the routing table for the router called Router4. This is important both following the formation of routing table in the execution mode. Packets are modeled as a DEVS messages and categorized as control and data packets. Control packets can carry bee agents.

3.2. Honeybees and Network Conceptual Models of Beehive

In this work, a swarm routing approach derived from honeybees and their interactions was developed using the concept of the honeybee scout-recruit mechanism during foraging. The starting point for developing the biologically inspired approach for modeling and simulating network systems is that “real bees can find optimum solutions in their foraging activity” (Seely 1995). Table 1 shows the analogy between honeybee colonies and computer networks.



IP Address: 0.0.0.4
phase: idle
sigma: infinity
tL: 0.020
tN: ∞
ReceiverQueueState: 0 packets(0KB)
sender queue length: 0
Routing Table of Router4

	0.0.0.3	0.0.0.2
0.0.0.3	0.67-NIC2	0.33-NIC1
0.0.0.2	0.33-NIC2	0.67-NIC1
0.0.0.1	0.40-NIC2	0.60-NIC1

Number of routes: 6
Global Routing Table of Router4null
Number of routes: 0
# of processed packets: 0

Figure 3: A Routing Table view in DEVJAVA, rows corresponds to destinations and columns to neighbors.

Computer networks correspond to the colonies of honeybees whose goal is to find paths to profitable nectar sources. Each network node is analogous to a beehive. The network links correspond to the scout-recruit system where honeybees leave their hives to gather food. Network nodes exchange control and data packets. Data packets correspond to foragers carrying nectar to beehives. The router inside each node uses the information received from all the nodes in the network (obtained by its control packets) to calculate the shortest path to each destination in terms of a chosen metric. The network links are used to limit the number of control and data packets that can be sent and received between nodes. This abstraction corresponds to the amount of information scouts and foragers can carry while searching for and transporting nectar. Each network node deploys a number of control packets (scouts) to find the most profitable paths for a given destination.

Table 1: Analogy between Network Systems and Honeybee Colonies

Computer Networks	Honeybee Colonies
nodes	beehive
network	nectar collecting area
spare network resources	nectar
links	flying to/from nectar/hive
control packets	scouts
data packets	foragers
routing information	dances and cues
Inter-network control packets	drones

The BEE (scout-recruit) routing algorithm is defined by the following rules:

Rule 1: Periodically or in event-triggered way, each node dispatches scout bees for gathering information about network for choosing the best route for sending packets over the network as in scouting the nectar during foraging.
Rule 2: During foraging, the goal of each scout is to collect nectar as much as possible.
Rule 3: Scouts then wonder around the network and gather information about the status of the network.
Rule 4: Given each node and its routing table, scouts choose neighbor nodes with estimated probabilities.
Rule 5: In the network, the nectar quantity can be collected by walking along a certain route is inversely proportional to the route cost.
Rule 6: Scouts are delayed at congested links.
Rule 7: Once a scout has reached its destination, it goes back to its source. During going back, it has high queuing precedence by which delay on links is kept less.
Rule 8: A scout never visits a node twice. To do so, scouts save a tabu list which includes the IPs of the visited node.
Rule 9: A scout never uses a link that does not have enough bandwidth.
Rule 10: A scout dies after it had reached its maximum number of hops.
Rule 11: When all or some scouts have returned to their beehives, the costs of found or recorded paths are evaluated and entered in routing table.

Self-organization of artificial bees is based on these relatively simple rules for derived from individual insect's behavior. These artificial bees correspond to a special class of automata called scout-recruit system that react to their local perception of the environment by stochastically adopting predefined behaviors. Autonomous actions are committee depending on local information and local interactions.

Control packets help control congestion by making alterations to routing tables in order to route new traffic away from congested nodes. Control packets also play a crucial role in ensuring survivability of the network based on the same principle honeybee colonies maintain their existence – i.e., by using scouts to search for nectar. Furthermore, the movement of control packets is used to balance network loads. Therefore, each node is capable of accommodating an ensemble of scouts for controlling network congestion.

4. SIMULATION MODELS

To show the capability (applicability) of the biologically inspired network system modeling approach, we started with well-known routing algorithms which are also used implemented. For instance, static link state algorithm is modeled to initialize network and distance vectors to calculate distances between nodes. In the implementation of all these algorithms, we used hop count as a metric although other metrics such as available link bandwidth may also be used.

Since one of the main objectives of this approach was to test performance of the new routing approach against the state-of-the-art algorithms, a representative network was used (see Figure 4). As we will show in the other example applications, by changing the number of node and link models and varying their parameters, alternative network topologies can be readily created. The network model used for comparing the above algorithms has 11 nodes (i.e., n1, ... , n11) and 18 bidirectional links (link1, ..., link18) – this is the representative network.

Each simulation run consisted of an adaptation to topology phase (initialization) and a test phase. During the initialization phase, system runs without load and initial routing tables are formed according to the number of hops (i.e., Dijkstra shortest path estimation algorithm (Dijkstra 1959)). During the test phase, the network performance was measured and recorded in terms of average packet delay, throughput, convergence time, and packet loss ratio. In Table 2, simulation parameters are summarized. All values are chosen according to algorithm test framework in which a representative network is used to see algorithm behavior and compare with other routing algorithms In the following section, in order to test algorithm across weighted conditions, all parameters are incremented.

Table 2: Simulation Model Parameters

Simulation Model Parameters	
Topology	11 routers, 18 bidirectional links
Simulation time	1 sec.
Node's buffers	1 MB
Node processor speed	1 msec/event
Link bandwidths	1.5 to 6 Mbps
Link delays	1 to 5 msec.
Traffic type	Uniformly random
Event frequency	1000 event / sec.
Packet sizes	10 to 100 KB

We used two standard performance metrics: throughput and packet delay. The amount of network traffic is determined by the number of packets in the network. Generally, many packets must wait in limited capacity (FIFO queue) for processing at the nodes. We avoided generation of packets with the same source and destination nodes, although this can be done as long as there is no direct source to destination connectivity (i.e., at least one link is used between the source and destination nodes).

Simulation Results & Discussion: We compare our approach with the RIP (Routing Information Protocol) algorithm which is commonly used in today's Internet. The BEE routing algorithm is shown to yield approximately 15.94% better throughput compared with

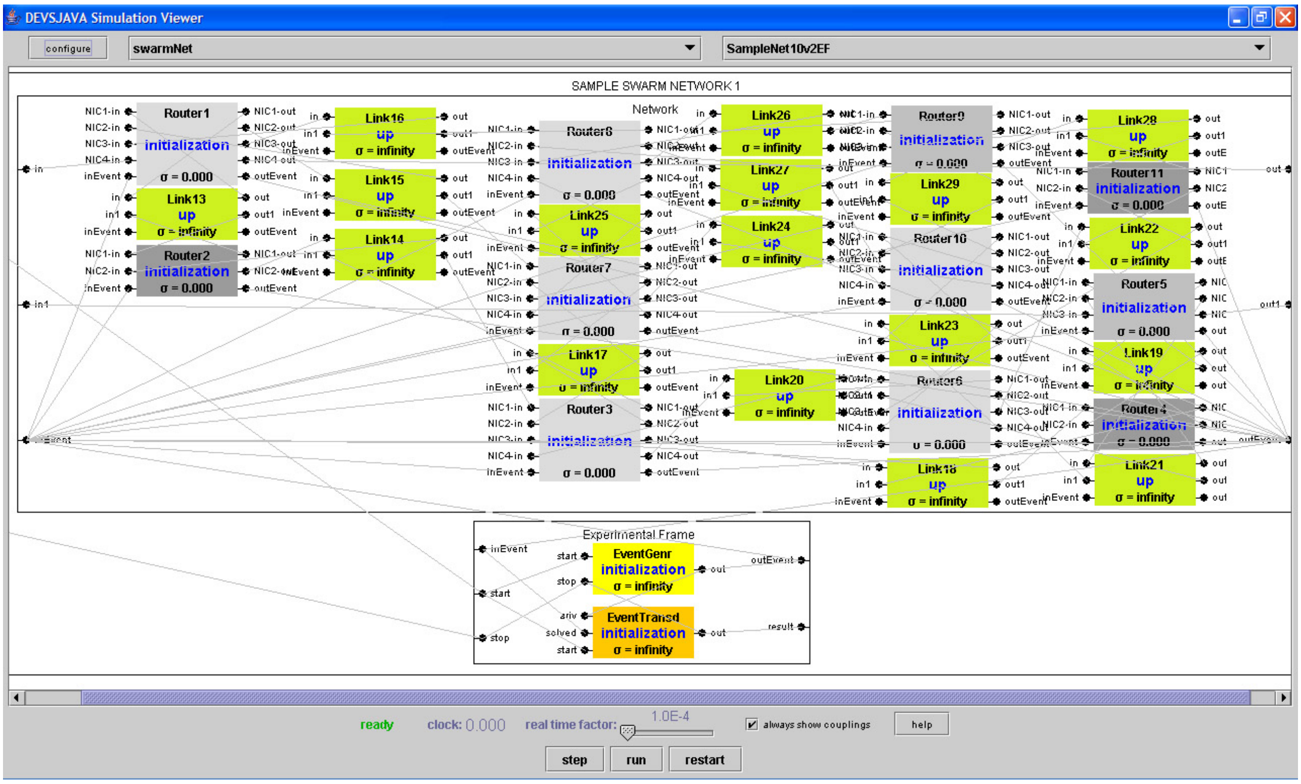


Figure 4: An Experimental Frame connected with its Network Model under DEVSJAVA

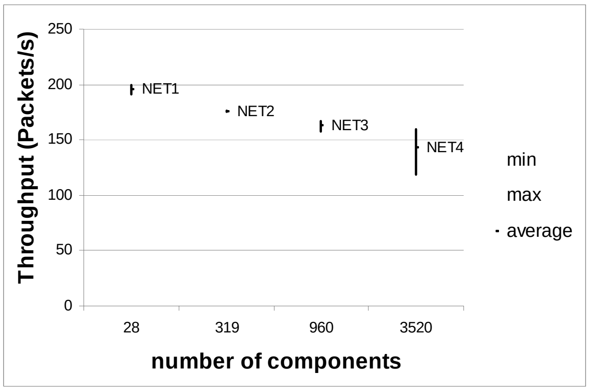
the RIP under optimum traffic load balancing (i.e., the BEE algorithm handles 916.34 packets/sec while the RIP algorithm can handle 790.5 packets/sec). Since throughput in the BEE algorithm reaches its maximum value in a shorter time (200 milliseconds), the BEE algorithm has better response time compared with the RIP algorithm. Once the network model with the BEE algorithm reaches the steady state throughput, the throughput remains nearly constant to the end of the simulation. In comparison, the network model with the RIP algorithm reaches its maximum throughput at a much later time (500 milliseconds). Therefore, the load balancing provided by the BEE algorithm is reached rapidly and evenly in the presence of heavy network traffic conditions.

We also compared the response times for each of these algorithms based on the turnaround time of the packets that are transmitted through the network. In our implementation, the turnaround time is defined as a packet's life-span time which starts from the packet generator and ends at the packet transducer while going through the nodes and links of the network mode. Average packet delays for the BEE and RIP algorithms are 7 and 9 milliseconds, respectively. The primary reason for this difference is attributed to the scouts in the BEE algorithm. The scouts find optimum routes quickly and thus allow faster response time to network changes. Bees keep the traffic low relatively to RIP. The ecological approach has better load balancing since the probabilistic routing used in the BEE algorithm forwards the packets' alternative routes. Furthermore, the network resources are better utilized and thus the network traffic load is distributed evenly across nodes and links. This reduces network congestions and results in the packets reaching their destinations faster. The simulation experiments show improved precision, stabilization and consistency of the BEE routing scheme. Also, the use of random traveling of the agents (scouts) increases the robustness of the network operation.

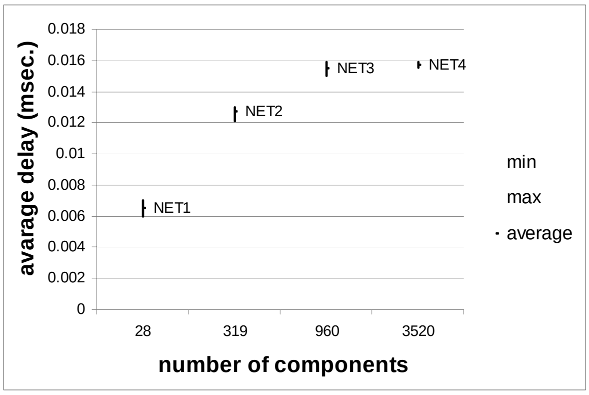
The developed approach supports modeling and simulating adaptive, robust, and survivable network applications. Since the SwarmNet environment is developed using the DEVS formalism and it supports modeling of networks using the biologically derived rules instead of complex formulas, simulation models can be developed systematically and simulated efficiently. Given that the need to better understand the Internet characteristics in terms of its topology, alternative configurations, and unpredictability of network traffic, researchers continue to develop greater capabilities to simulate large-scale models (Fujimoto et al. 2003, Floyd and Paxon 2001). For example, simulations having more than 100,000 routers and nodes have been developed using dozens of parallel processors (Riley, Fujimoto, and Ammar 1999; Zegura, Calvert, and Bhattacharjee 1996; Cowie, Nicol, and Ogielski 1999).

Simulation experiments of large-scale network models: A primary benefit of a network-based modeling approach is its degree of support for large-scale model development and efficient simulation. Due to both scale and complexity of current network systems such as the Internet, modeling and simulation of these systems is non-trivial (Floyd and Paxson 2001). While scalability issue is due to the routing databases of the nodes increasing with the network size, which can cause some routers' databases to exceed their capacities, complexity comes from variety of communication media, communications equipment, protocols, and hardware and software platforms found in the network. To allow simple redesign of the routing database for large-scale networks, the above clustering approach was developed.

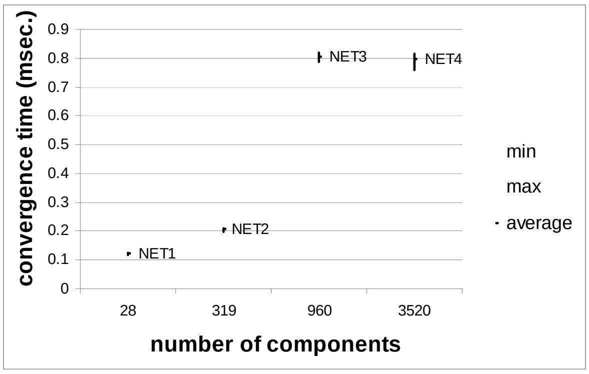
In order to study the scalability of the proposed approach, we developed models for networks ranging from 29 to 3520 components. Small networks were created manually while large networks were produced using a recursive topology-generating algorithm. To verify and validate the approach on larger models, a set of experiments were carried out and the results have been evaluated in a comparative manner. In these experiments, the key independent variables are the degree of network connectivity and the number of network components. Networks were modeled and their simulation results were analyzed. The relation between the network throughput and the number of nodes is shown in Figure 5(a). The throughput gradually decreases as the number of components increases since the packet loss ratio increases in accordance with the size of the models. However, performance losses for large networks remain acceptable. Another observation is that for all network sizes, the average delay across the networks is increasing but not asymptotically increasing (see Figure 5(b)).



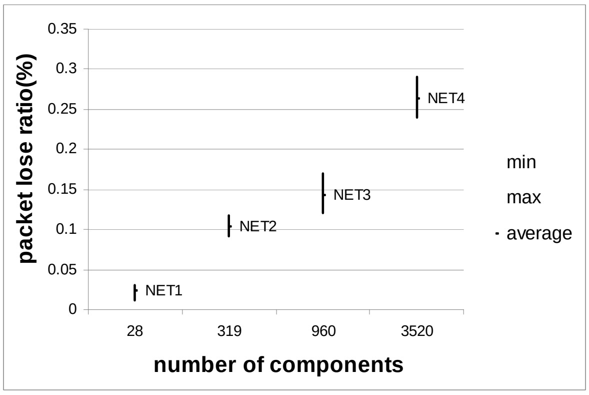
(a)



(b)



(c)



(d)

Figure 5: Large-scale network (a) throughput, (b) average delay, (c) convergence time, and (d) packet loss measurements using the BEE algorithm.

Rapid convergence is a main feature of any efficient routing algorithm. A routing algorithm must show how quickly it can construct and update the nodes' routing tables given different scales of networks. Figure 5(c) shows that the convergence time of the biologically inspired routing algorithm is scalable. Although larger networks exhibit a relatively long time to converge as compared with smaller sized networks, their convergence times are in milliseconds. A stationary trait can be recognized in the convergence trajectory as the scale of the network approaches a thousand components. The reason is attributed to the network being composed of similar networks since larger models are recursively and automatically constructed. This approach together with the parallelism in the DEVSJAVA simulation engine causes convergence time to increase less while the number of components increases.

Finally, as shown in Figure 5(d), the packet loss ratio gradually increases with the increase in the number of components. The packet loss is shown to be linear or better as the scale of the network is increased. Simulations were executed in the DEVSJAVA environment for a period of ten seconds. Using a Windows computer with 2.4GHz processor and 512M RAM, the representative network took a few minutes to execute whereas the largest network simulation took less than three hours to complete.

5. CONCLUSIONS

In this paper, we incorporated biologically inspired modeling constructs into the general-purpose DEVS modeling framework. The resulting SwarmNet modeling approach affords scalable and efficient simulation of computer network systems. The developed BEE routing algorithm derived from the concepts found in the social insect societies show better performance compared with the commonly used RIP algorithm. The proposed approach shows better response time for discovery and deployment of new routes and affords higher robustness. The use of the control (scouts) packets does not play a significant role in the total network load due to their lightweight design. In the case of network malfunction such as link unavailability or node congestion (i.e., node has reached the maximum number of packets it can process), the network remains functional since the probabilistic routing adapts faster to fluctuations in the network and can find alternative paths for destinations at run-time. Based on these observations, the network has higher survivability against surges.

From the perspective of model specification, the node and link models can be extended to use probabilistic timing and include security features. From the application vantage point, it would be interesting to apply this approach to modeling crowd behaviors since existing simulation environments lack the underlying formal theory provided by DEVS. Finally, this SwarmNet modeling approach can support design of emergent and scalable network systems and can be simulated in distributed and/or service-oriented computing technologies.

REFERENCES

ACIMS. Arizona Center for Integrative Modeling and Simulation. 2008. http://www.acims.arizona.edu/SOFTWARE/software.shtml

Bonabeau, E., Dorigo, M., and Théraulaz, G., 1999, *Swarm Intelligence: from natural to artificial systems*, Oxford University Press.

Cowie, D., Nicol, M., and Ogielski, A. T., 1999. Modeling the global internet. *Computing in Science and Engineering*, vol. 1, no. 1, pp. 42–50.

Dijkstra, E.W., 1959. A Note on Two Problems in Connexion with Graphs. *Numerische Mathematik* Vol. 1.

Floyd and Paxson, V., 2001. Difficulties in Simulating the Internet, *IEEE/ACM Transactions on Networking*, vol. 9, no. 4, pp. 392–403.

Fujimoto, R., Perumalla, K., Park, A., Wu, H., Ammar, M., and Riley, G., 2003. Large-scale network simulation: how big? how fast?. *11th IEEE/ACM International Symposium on Modeling, Analysis and Simulation of Computer Telecommunications Systems(MASCOTS)*, p. 116, 2003.

Lunceford, W.H. and Page, E.H., 2002. *Grand Challenges for Modeling and Simulation*, Editors, Western Multiconference, San Antonio, TX.

Riley, G., Fujimoto, R., and Ammar, M. H., 1999. A generic framework for parallelization of network simulations, simulation: how big? how fast?. *11th IEEE/ACM International Symposium on Modeling, Analysis and Simulation of Computer Telecommunications Systems(MASCOTS)*, pp. 128–, 1999.

Sarjoughian, H., and Cellier, F., 2001. *Discrete Event Modeling & Simulation Technologies: A Tapestry of Systems and AI-based Theories and Methodologies for Modeling and Simulation.* Springer Verlag, 2001.

Seely, T.D., 1995. *The Wisdom of the Hive*. Cambridge, Mass: Harvard University Press.

Steenstrup, M. E. (Ed.)., 1995. *Routing in Communications Network.* Prentice-Hall.

Wymore, W.A. 1993. *Model-based Systems Engineering: An Introduction to the Mathematical Theory of Discrete Systems and to the Tricotyledon Theory of System Design*, Boca Raton, CRC.

Zegura, Calvert, K. L., and Bhattacharjee, S., 1996. How to model an internetwork, *IEEE INFOCOM*, vol. 2. pp. 594–602, March 1996, San Francisco, CA

Zeigler, B.P., Praehofer, H., and Kim, T.G., 2000. *Theory of Modeling and Simulation: Integrating Discrete Event and Continuous Complex Dynamic Systems*. Second Edition Academic Press.

Zeigler, B.P., Sarjoughian, H.S., 2003. *Introduction to DEVS Modeling & Simulation with JAVA: Developing Component-based Simulation Models*, Available from: http:// www.acims.arizona.edu/PUBLICATIONS.

Zeigler, B.P., Mittal, S., 2001. Modeling and Simulation of Ultra-large Networks: A Framework for New Research Directions. *ULN Workshop*, July 2002.

Zengin, A., Sarjoughian, H.S, and Ekiz, H., 2004. Biologically Inspired Discrete-Event Network Modeling. *Proceedings of the European Simulation Symposium*, pp. 317-324, Budapest, Hungary, Oct. 17-20.

AUTHORS BIOGRAPHY

AHMET ZENGIN is Assistant Professor at Sakarya University, Turkey. His experience with modeling and simulation includes a one-year-stay in ACIMS Lab at the Arizona State University. His research topics include DEVS theory, multi-formalism modeling, parallel and distributed simulation, modeling and simulation of large-scale networks, distributed systems management, biologically-inspired optimization schemes. His main research interest lies in parallel and distributed simulation and the High Level Architecture.

HESSAM S. SARJOUGHIAN is Assistant Professor of Computer Science and Engineering at Arizona State

University, Tempe and Co-Director of the Arizona Center for Integrative Modeling and Simulation. His research includes modeling theory, multi-formalism modeling, collaborative modeling, distributed co-design, intelligent agents, and software architecture. His professional experience has been with Honeywell and IBM. Visit <http://www.eas.asu.edu/~hsarjou> and <http://www.acims.arizona.edu> for more information.

HUSEYIN EKIZ is received M.Sc. in 1993 from Gazi University, Turkey, and Ph.D. degree in computer engineering in 1998 from the University of Sussex, England. He is currently Professor of the Department of Computer Systems Education and Dean of the Technical Education Faculty, Sakarya University, Turkey. His research interests are in the fields of network systems, distance education, digital circuit design and microprocessor architectures.

EMERGENCE, ANTICIPATION AND MULTISIMULATION: BASES FOR CONFLICT SIMULATION

Tuncer Ören[a], Francesco Longo[b]

[a]M&SNet (McLeod Modeling and Simulation Network)
SITE, University of Ottawa, Ottawa, ON, Canada

[b]M&SNet (McLeod Modeling & Simulation Network)
(MSC-LES) Modeling & Simulation Center - Laboratory of Enterprise Solutions
University of Calabria, Italy

[a]oren@site.uottawa.ca, [b]f.longo@unical.it,

ABSTRACT

Two major categories of usages of simulation are highlighted. They are (1) providing experience for three types of training and entertainment and (2) performing experiments. Several characteristics of emergence are reviewed, especially within the framework of the simulation of social systems: and lists of over 30 terms related with emergence as well as over 20 types of emergences are given as appendices. The importance of perception in human decision making and activities is underlined; anticipation and early detection of emergence are presented as a special cases of perception. Basis of a methodology to model emergence is explained.

Keywords: Emerge, de-emerge, re-emerge, merge, demerge, multistage model, multisimulation, types of simulation, perception, anticipation, behaviorally anticipatory systems, simulation challenge, conflict simulation

1. INTRODUCTION

Our long term interests include the development of appropriate modeling and simulation methodologies for conflict studies applicable in different types of usages of simulation. To better represent several human traits which can be best described by mostly linguistic variables, such as autonomy/quasi-autonomy, personality and emotions, agent simulation is preferred. The focus in this article is on perception, anticipation, and emergence within the framework of multisimulation. A modeling methodology to represent emergence is proposed and multisimulation is explained as a simulation methodology to allow simulation of several aspects of a system of interest.

2. SIMULATION

2.1. Types of Simulation

Simulation is useful for two major categories of usages which are (1) providing experience for three types of training and entertainment and (2) performing experiments.

In *training*, simulation is use of a representation of a system to gain/enhance competence through experience under controlled conditions. Training can be achieved (1) by using virtual equipment as it is the case in simulators and virtual simulators (i.e., virtual simulation) to gain/enhance motor skills, (2) by gaming simulation (i.e., constructive simulation to gain/enhance decision making skills, or (3) by a mixture of real system and simulation (i.e., live simulation) to gain/enhance operational skills.

In areas other than training and entertainment, simulation is used to perform goal-directed *experiments* with dynamic models. These areas include education, understanding, and decision support.

Use of simulation for decision support is done for the following main categories of activities:

- *Prediction* of behavior or performance of the system of interest within the constraints inherent in the simulation model (e.g., granularity) and experimental conditions;
- *Evaluation of alternative* models, parameters, experimental and/or operating conditions on model behavior or performance;
- *Sensitivity analysis*;
- *Engineering design*;
- *Prototyping*;
- *Planning*;
- *Acquisition* (or simulation-based acquisition);
- *Proof of concept.*

The richness of the discipline of simulation and its many facets are elaborated on in other articles, see for example Ören (2005, 2006, 2009). However, some authors, instead of benefiting from these scientific aspects of simulation which make simulation a vital enabling technology for many disciplines–including social sciences– they tend to focus either (1) on the non-technical historic meaning of the term simulation which exists in English since the 14th century or (2) on an assumption not done by simulationists.

2.2. Simulation Fallacy

In a review of a book by Buchanan (2007a, b), Feld refers to "Simulation fallacy" in the following assertion: "Furthermore, in searching for explanations for social phenomena, Buchanan repeatedly commits two types of logical errors. The first of these is the simulation fallacy. A simulation typically shows that a particular proposed mechanism "could" produce an observed pattern. However, a simulation provides no evidence that the proposed mechanism is necessarily important or even relevant in some particular situation in which the pattern is found. Identifying "possible" causes only contributes to a scientific understanding of society when the conditions under which those causes are applicable and the extent of their applicability are empirically demonstrated and understood."

The term "*simulation fallacy*" is also used to stress the fact that the simuland (the system which is simulated) is not the same as the model which is used in its simulation. This assumption is categorically not done in either type of simulation stated at the beginning of the section on the types and usages of simulation, namely in (1) providing experience for three types of training or entertainment and (2) performing experiments.

"Simulation fallacy" is also stated within a metaphysical framework, as expressed by Searle: "Given that a computer simulation of a fire doesn't burn the neighbourhood down, or a simulation of gold make you rich, why should the computer simulation of understanding, actually understand?"

According to Copeland, the term simulation fallacy is also stated as:" A closely related error, unfortunately also common in modern writing on computation and the brain, is to hold that Turing's results somehow entail that the brain, and indeed any biological or physical system whatever, can be *simulated* by a Turing machine."

2.3. Simulation Challenge

One can take the following quotations–all published in 1968–as a "*simulation challenge*." First, a quotation from Knuth (1968): "*We often fail to realize how little we know about a thing until we attempt to simulate it on a computer.* "

Second, two observations are from John McLeod (1968): *"(1) If you do not know enough about a system, a good way to find out more is to try to simulate it. And (2) A model need not be sophisticated to be useful. If it simulates those aspects of interest to the degree necessary for the study at hand, the simulation is valid*". And the last quotation which is also from John McLeod (1968): "*The smaller a man, the closer his horizons*".

2.4. Agent Simulation

"Agents are autonomous software modules with perception and social ability to perform goal-directed knowledge processing, over time, on behalf of humans or other agents in software and physical environments. The knowledge processing abilities of agents include: reasoning, motivation, planning, and decision making. Additional abilities of agents are needed to make them Intelligent, human-like, and trustworthy. Abilities to make agents intelligent include anticipation, understanding, learning, and communication in natural language. Abilities to make agents more trustworthy as well as assuring the sustainability of agent societies include being rational, responsible, and accountable. These lead to rationality, skillfulness and morality (e.g., ethical agent, moral agent)." (Ghasem-Aghaee and Ören, 2003).

Abilities to make agents human-like include representation of personality, emotions, and culture.

The term *agent simulation* denotes simulation of systems modeled as software agents.

3. EMERGENCE

3.1. Sistemic View

"It is important to draw the line between "emergence" as a mere synonym of everyday language words like "appearance" or "growth" on the one hand , and 'emergence' as the fundamental concept of emergentist theories in philosophy on the other hand." (Brunner and Klauninger).

In philosophy as well as in systems science, "emerge" and its derivatives have different meanings and these properties are very important in complex systems in general and in the simulation of social systems in particular; therefore they are elaborated in this article. For a history of emergent properties, see Stanford Encyclopedia of Philosophy (SEP-emergent). A systemic definition of emergence is given by Goldstein (1999) as: "the arising of novel and coherent structures, patterns and properties during the process of self-organization of complex systems."

Some fundamental concepts are:

- Emergence is a fundamental feature of self-organizing systems;
- Self-organizing systems are complex systems;
- Complex systems are not organized centrally;
- The complexity of a system depends on the number of its elements and connections between the elements;
- Emergence is based on non-linear causality.

There are feedback loops within a self-organizing system. Fenzl elaborates on the role of energy flow and information in emergence and self-organization of complex systems (Fenzl). "In all self-organizing (physical, biological, and social) systems the emergence of order is triggered by fluctuations that cause synergies between the elements of the systems. … There is a non-linear, complex relationship between causes and effects in self-organizing systems: It is objectively conditioned that a fluctuation will at some critical point in the system´s development result in the emergence of new order (necessity). According to Fuchs, but the exact moment and the exact form of the process of emergence and its resulting new qualities is to a large degree uncertain (chance)."

Fuchs lists and clarifies 15 principles of physical self organization.

These principles are listed in Table 1.

Table 1: Principles of Physical Self-Organization

1. Control parameters
2. Critical values
3. Fluctuation and intensification
4. Feedback loops, circular causality
5. Non-linearity
6. Bifurcation points
7. Selection
8. Emergence of order
9. Information production
10. Fault tolerance
11. Openness
12. Symmetry breaking
13. Inner conditionality
14. Relative chance
15. Complexity

Appendices 1 and 2 adopted from Fuchs (pp. 206-207) and (Arshinov and Fuchs, pp. 6-8) cover some explanations of these fundamental concepts as well as types of emergences.

Bonabeau et al. (1995) provide a critical review of emergent phenomena. Deguet et al. (2006) provide a survey of definitions of emergence. A special issue of Sciences et Avenir is dedicated to discuss several aspects of emergence (S&A, 2005).

3.2. Types of Emergence: some lexical clarifications

Rey (2005) provides a good etymological clarification for the relationships of the terms emergence, immersion, and submersion. In our article, the relevance of the terms merging, de-emergence and re-emergence are also stressed since they are equally relevant in simulation of social studies, especially for conflict studies. Table 2 depicts the relationships of the terms immerge, submerge, emergence, re-emergence, de-emergence, merging (horizontal, vertical), and demerging.

Table 2. Relationships of the Terms related with "merge"

⤈	—— ↓	—↑—	—— ↓	↔ ↕
Immerge immerse	Subm.	e(x)m. Re-em.	de-em.	Merge Hor.-m. Ver.- m. Dem.

Immerge, submerge, emerge, de-emerge, re-emerge, and merge are derived from Latin *emergere* which means to dip.

Original connotations were passing–totally or partially–from a medium (air) to another (water) as it is the case in *immerge* (from Latin *in-* and *mergere* "to plunge") and *submerge* (from Latin *sub-* "under"). *Submerge* means:

- to place under water;
- to cover with water, inundate;
- to hide from view, obscure.

Immerge (immerse) means:

- to plunge into something that surrounds or covers, *especially***:** to plunge or dip into a fluid.
- to engage wholly or deeply, absorb: *scholars who immerse themselves in their subjects* (AHD-emerge).

In the case of emerge (from Latin ex- "out of") the direction of the flow is reversed and original meaning was passing from a medium (water) to another (air). Later, immerge, submerge, and emerge gained additional meanings.

The term "*emergence*" can be used with its system theoretic connotation meaning appearance of a characteristic of a non-linear complex system with feedback loop(s). In social systems this feature is of primordial importance.

As it is the case with most words in natural languages, the word "emerge" and its derivatives have different meanings and some of them differ from their etymological meanings.

For example, according to the American Heritage dictionary, "*emerge*" means the following:

- to rise from or as if from immersion: Sea mammals must emerge periodically to breathe;
- to come forth from obscurity: new leaders who may emerge;
- to become evident: The truth emerged at the inquest.
- to come into existence (TAH-emerge).

According the Merriam-Webster, "emerge" means:

- to become manifest, become known: new problems *emerged*;
- to rise from or as if from an enveloping fluid, come out into view: a diver *emerging* from the water;
- to rise from an obscure or inferior position or condition: someone must *emerge* as a leader;
- to come into being through evolution (MW-emerge).

Compact Oxford Dictionary defines “emerge” as follows:

- become gradually visible or apparent;
- (of facts) become known;
- recover from or survive a difficult period (OD-emerge).

The term "*merge*" means:

- to cause to be absorbed, especially in gradual stages;
- to combine or unite: *merging two sets of data*;
- to blend together, especially in gradual stages;
- to become combined or united.

Merging, hence, mergence may end up of generating characteristics and/or structures different then previous ones and therefore is an important concept in social system simulation.

Demerge (Brit.) is to separate a company from another which was merged.

In social systems, two types of merging, namely horizontal merging and vertical merging can cause structural changes in system components. *Horizontal merger* is a merger occurring between companies producing similar goods or offering similar services. This business term can be extended to factions of similar goals. *Vertical merger* is a synonym of vertical integration which is defined as follows by Business Dictionary (BD-vi): "*Merger of firms at different stages of production and/or distribution in the same industry. When a firm acquires its input supplier it is called backward integration, when it acquires firms in its output distribution chain it is called forward integration. For example, a vertically integrated oil firm may end up owning oilfields, refineries, tankers, trucks, and gas (petrol) filling stations. Also called vertical merger.*" In conflict situations, vertical merger may mean merging of factions to complement their functions.

3.3. De-emergence and Re-emergence

In social systems, the opposite characteristic of emergence is equally important; namely a characteristic or a component of the system may disappear. Instead of using the terms immersion or submersion, the term de-emergence is preferred; since the Latin prefix *de-* also means negative and opposite. Another relevant topic is re-emergence, i.e., emergence which may occur after de-emergence.

3.4. Emergence in Simulation of Social Systems

Jin et al. (2008) provide a review of emergence-oriented research in agent systems. Dessalles et al. (2008) elaborate on emergence in agent based computational social science. Different types of emergence, i.e., emergence, de-emergence, re-emergence, horizontal and vertical merging are important in the simulation studies of conflict problems in social systems. Sawyer (2004), for example. focuses on emergence in agent simulation of social systems and simulates the mechanisms of emergence. Sawyer also addresses the question "Which social properties are emergent?"

In social system dynamics, several aspects of emergence need to be taken into account:

- Emergence studied *as a systemic characteristic of non-linear complex systems*; since social systems are both complex and non-linear;
- in addition to self-organization as a complex system, *a social system may be guided (or forced) to change* (for the good of the citizens or for the group who may have an interest for the change). This way, external and internal inputs/events/ processes affect the social behavior. For example, conditions may be activated to cause, in the long run, a new feature in the society. This new feature may change the nature of the society even radically and may appear to have emerged out of nowhere. However, if the citizens and/or those working for them (e.g., representatives and media) would be proactive and anticipate the consequences, they could have foreseen that the initial events/processes/inputs would be resulting with such effects. Most often, the change occurs at several stages. Hence, in social systems, leading to conflict, especially the first two principles of self-organization as listed by Fuchs may be useful in simulation modeling. They are: "(*1) control parameters*: a set a parameters that influence the state and behavior of the system. (*2) critical values*: if certain critical values of the control parameters are reached, structural change takes place, the system enters a phase of instability/criticality (Fuchs, pp. 206-207).
- Also, *disappearance* of some social characteristics may also be conceived and hence, modeled and simulated as "emergence."

4. PERCEPTION, ANTICIPATION AND BEHAVIORALLY ANTICIPATORY SYSTEMS

4.1. Perception

Perception is very important in our activities; since the way we perceive reality affects our decisions, feelings, emotions, and activities. We can perceive three categories of entities which exist, which may be anticipated, and which may emerge or which may cease to exist. In modeling existing, anticipated, and emergent systems, we represent goal-directed abstractions of objects, their attributes, and relationships among themselves as well as with their environments.

4.2. Anticipation and Anticipatory systems

Most human activities and simulation models are reactive; therefore according to causality, the value of the current input and current state determine the value of the next state. Special Interest Group in Anticipatory Systems of BISC (Berkley Initiative in Soft Computing) lists 12 definitions of anticipation. However, the following definition, listed as the first one is the essence of anticipatory systems: “An anticipatory system is a system whose current state is determined by a future state. The cause lies in the future.” (Rosen, 1985) (BISC-SIG-AS).

An anticipatory system may be difficult to model, since the value of its current state would be determined by the value of a future state. However, by defining a behaviorally anticipatory system, one can circumvent this problem as follows: a behaviorally anticipatory system can have, at time t, one or more predicted model(s) of itself and/or of its environment at time *t+n*. Hence, at time *t*, its next state can depend on the current image of the predicted model (at *t+n*) of its state and/or its environment. Since, current image, at time *t*, of the predicted model of itself and/or of it environment is well specified, next state can be calculated without any contradiction with causality.

Ören and Yilmaz (2004):

1. give a systematic and comprehensive classification of input and point out the relevance of perception as an important type of input in intelligent systems;
2. give a categorization of perception and present anticipation as a type of perception;
3. clarify the inclusion of anticipation in simulation studies and elaborate on other aspects of perceptions in simulation studies especially in conflict situations.

5. MULTISTAGE MODELS AND MULTISIMULATION

The sequence of models with emergent states and/or transitions can be conceived as a set of multistage models which can then be simulated by multisimulation. The concepts of multistage models and multisimulation were introduced in 2001 (Ören, 2001) as a methodology to allow simulation of several aspects of a system simultaneously, especially for conflict management studies. Elaborations on multisimulation are made by Yilmaz and Ören (2004), Yilmaz et al. (2006); Yilmaz et al. (2007), Lim (2007) and Yilmaz and Tolk (2007).

In some aspects of social system dynamics, completely new conditions may emerge and accordingly, a simulation study would need to be interrupted, model needs to be replaced by a new one and then the simulation study would resume. In some cases, at the update instant of the simulation study, one may want to continue with two or more models under same or different experimental conditions.

This concept would lead us to multistage modeling and multisimulation since after the interrupt, more than one simulation study (successor simulation study) would occur. If the successor simulations are realized under the same experimental conditions, they may be coupled or not. Coupled successor simulations can have common resources.

Figure 1, taken from Ören (2001), depicts an example multistage model.

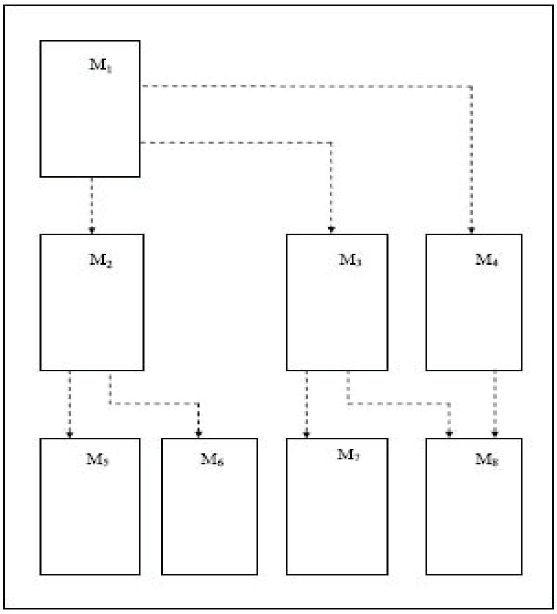


Figure 1: An Example Multistage Model

In the example, the following multistage models are identified: M1, M1M2, M1M2M5, M1M2M6, M1M3, M1M3M7, M1M3M8, M1M4 and M1M4M8. A model, which can be used after another one in a multistage model, is a successor model. Normally all the multistage models may not be known a priori. Only the initial model M1 may be known. In this case, one can attempt to model alternative models to get ready for contingencies. Supposing that M2 and M3 are also modeled, one can have two multistage models: M1M2 and M1M3. One can perform a simulation study with each multistage model to find out for example, the outcomes of having M2 or M3. Accordingly, one can try to control the conditions to facilitate transition to a specific model module and/or to make it difficult the transition to another one. If the status of a module of a multistage model is not acceptable or desirable, one has to generate successor model(s) and facilitate transition to that module model,(Ören, 2001).

6. BASES FOR A MODELING METHODOLOGY: EMERGENCE OF TRANSITIONS AND STATES

Figure 2 depicts emerged states (g) and (h) and emerged transitions a-c, d-g, and g-f, h-f, and g-h. In Figure 2, a dynamic system is represented as a state machine. To make the point of two types emergences, let's consider that initially, the system consists of the states (a) through (f), the initial state is (a), and that only transitions represented by solid arrows exist. The two types of emergences that we would like to elaborate on are:

1. *Emergent transitions*: Under certain conditions of the control parameters (or control variables) and their critical values (or threshold values), a transition may emerge, for example from state (a) to state (c) as represented in Figure 2. Similarly, under certain conditions of control variables and their threshold values, some transitions may *disappear* or may be *deactivated*. In the case of disappearance, the change is long-term (it can also be irreversible). In the case of deactivated transitions, the transition is reversible and depends on the occurrence of favorable conditions.
2. *Emergent states*: Under certain conditions of the control variables and their threshold values, a new state may emerge. In the example, state (g) emerges under favorable conditions. Emergence of a state, (h) for example, may start by emergence of a state, (g) in the example, and be realized by a series of emergent states. In evolutionary systems, this may correspond to the evolution of the system. Under other favorable conditions a state may disappear or may be deactivated. Once a state emerges, still under some favorable conditions transition to and transition from it may emerge.

Emergence of transitions and states may not be dichotomic; i.e., they can best be represented as fuzzy entities having a presence of 0% to 100%.

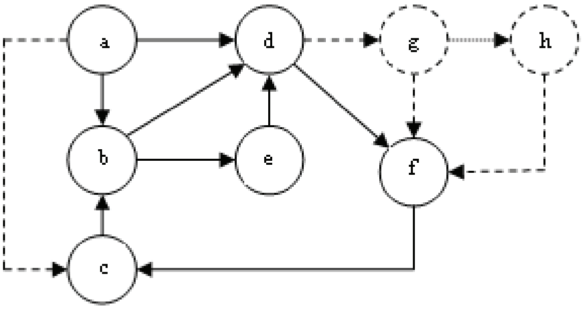


Figure 2. Emerged states and emerged transitions in a state machine

7. CONCLUSIONS

Both types of simulation are important in conflict studies to either provide experience under controlled conditions, hence for training decision makers; or for performing goal-directed experiments to analyze conflict situations to test hypotheses or to understand the mechanisms. In such studies being reactive is not sufficient; the needed proactivity can be modeled and simulated as a behaviorally anticipatory system. Furthermore, social systems, being highly non-linear complex systems can exhibit several types of emergence. Development of proper modeling formalisms for representing emergence and using multisimulation may be useful in the study of conflict problems. Representation of several types of emergence of states and transitions discussed in this article as well as multisimulation may be useful in simulation and management of conflicts.

APPENDIX A – TYPES OF EMERGENCE

- **Behavior emergence**;
- **Bottom-up emergence**: A perturbation causes the system parts to interact synergetically in such a way that at least one new feature on a higher level emerges (Arshinov and Fuchs, pp. 7- 8);
- **De-emergence**;
- **Emergence**: the appearance of a new property of a system which cannot be deduced or previously observed as a functional characteristic of the system. … Generally, higher level properties are regarded as emergent. For example, water has emergent properties different from its interconnected parts (molecules of H and O). These properties disappear if the molecules are separated again (Fenzl, pp. 252). The notion of emergence means that a system is more than the sum of its parts and that a developing system has new qualities that can't be reduced to old states or prior existing systems (Brunner and Klauninger, pp. 23).
- **Emergence of coherency between two open systems**;
- **Emergence of order**: in all self-organizing (physical, biological, and social) systems the emergence of order is triggered by fluctuations that cause synergies between the elements of the systems (Fuchs, pp. 219);
- **Epistemological emergence**;
- **Horizontal merger**;
- **Immersion**;
- **Innovative emergence**: the representation of attaining a new model class that improves the modelling process in the first place (Zimmerman, pp. 38).
- **Macro emergence**;
- **Merger**;
- **Micro emergence**;
- **Non-reflexive emergence**;
- **Ontological emergence**;
- **Re-emergence**;
- **Reflexive emergence**;
- **Second-order emergence**;
- **Semantic emergence**;
- **State emergence**;
- **Strong emergence**;
- **Structural emergence**;
- **Submersion**;
- **Top-down emergence**: downward causation can be described as top-down-emergence if

new qualities of certain parts (seen as wholes or systems themselves) show up (Arshinov and Fuchs, pp. 7).
- **Transition emergence**;
- **Vertical merger**;
- **Weak emergence**.

APPENDIX B – BASIC CONCEPTS RELATED WITH EMERGENCE

- **Cohesion**: cohesion means the closure of the causal relations among the dynamical parts of a dynamical particular that determine its resistance to external and internal fluctuations that might disrupt its integrity (Arshinov and Fuchs, pp. 7-8).
- **Complex**;
- **Complicated**;
- **Downward causation**: once new features of a system have emerged they along with the other structural macro-aspects of the system influence, i.e. enable and constrain, the behavior of the system parts. Downward causation means the localization of more global qualities (Arshinov & Fuchs, pp. 7-8);
- **Emerged**;
- **Emerged feature**;
- **Emergence condition**: necessary and sufficient condition to trigger an emergence;
- **Emergent**;
- **Emergent behavior**;
- **Emergent condition**;
- **Emergent dynamics**;
- **Emergent feature**;
- **Emergent function**;
- **Emergent functionality**;
- **Emergent input**;
- **Emergent output**;
- **Emergent output function**;
- **Emergent property**: there is the important assertion that some properties of the whole cannot be explained by, or deduced from, the properties of the parts. Such properties are emergent, as opposed to resultant properties;
- **Emergent state**;
- **Emergent structure**;
- **Emergent transition**;
- **Emergent transition function**;
- **Emergentism**;
- **Emergentist**;
- **Emerging feature**;
- **Emerging paradigm**;
- **Emerging phenomenon**;
- **Globalization and Localization**: bottom-up emergence means the globalizing sublation of local entities, downward causation the localization of more global qualities.
- **Hierarchy**: the *s*elf-organization of complex systems occurs in a hierarchical systems. Upper levels are more complex and have additional emergent features (Arshinov and Fuchs, pp. 6-7).
- **Macroscopic emergent behavior**;
- **Non-emergent**;
- **Non-emergent property**;
- **Non-linear causality**: causes and effects can´t be mapped linearly. Similar causes can have different effects and different causes similar effects; small changes of causes can have large effects whereas large changes can also only result in small effects but nonetheless it can also be the case that small causes have small effects and large causes large effects. Emergence is based on non-linear causality (Arshinov and Fuchs, pp. 7).
- **Ontology of emergence**;
- **Self organization**: self-organization means *appearance of new system structures* without explicit pressure from outside the system, or involvement from the environment. In other words, the constraints on the organization of the system are internal phenomena, resulting from the interactions among the components and usually independent of their physical nature. Self-organization can produce structural changes maintaining a stable mesoscopic form of the system, or show transient phenomena (Fenzl, pp. 252).
- **Self-organizing systems**: self-organising systems are complex systems. Emergence is a fundamental quality of self-organising systems. Self-organisation of complex systems produces a hierarchy in two distinctive senses: the level of emergence is a hierarchically higher level, i.e. it has additional, new emergent qualities that can´t be found on the lower level which is comprised by the components. The upper level is a sublation of the lower level; self-organisation results in an evolutionary hierarchy of different system types, these types are hierarchically ordered in the sense that upper levels are more complex and have additional emergent qualities (Arshinov and Fuchs, p. 6-7);
- **Semantic emergent behavior**;
- **Spatially emergent behavior**;
- **Syntactic emergent behavior**;
- **Systemness**: self-organisation takes place in a system, i.e. in coherent whole that has parts, interactions, structural relationships, behavior, state, and a border that delimits it from its environment (Arshinov and Fuchs, pp. 6-7).

APPENDIX C – PRINCIPLES OF PHYSICAL SELF-ORGANIZATION

(from Fuchs, pp. 206-207)

- *control parameters*: a set a parameters influences the state and behavior of the system;
- *critical values*: if certain critical values of the control parameters are reached, structural change takes place, the system enters a phase of instability /criticality;
- *fluctuation and intensification*: small disturbances from inside the system intensify themselves and initiate the formation of order;
- *feedback loops*: there are feedback loops within a self-organizing system;
- *non-linearity*: in a critical phase of a self-organizing systems, causes and effects can´t be mapped linearly: similar causes can have different effects and different causes similar effects; small changes of causes can have large effects whereas large changes can also only result in small effects (but nonetheless it can also be the case that small causes have small effects and large causes large effects);
- *bifurcation points*: once a fluctuation intensifies itself, the system enters a critical phase where its development is relatively open, certain possible paths of development emerge and the system has to make a choice. Bifurcation means a phase transition from stability to instability;
- *selection*: in a critical phase which can also be called point of bifurcation, a selection is made between one of several alternative paths of development;
- *emergence of order*: in a critical phase, new qualities of a self-organizing system emerge; this principle is also called order from chaos or order through fluctuation. A self-organizing system is more than the sum of its parts. The qualities that result from temporal and spatial differentiation of a system are not reduceable to the properties of the components of the systems, interactions between the components result in new properties of the system that can´t be fully predicted and can´t be found in the qualities of the components. Microscopic interactions result in new qualities on the macroscopic level of the system;
- *information production*: new features (in the original text the term "quality" is used. Instead we used the term "feature" to be neutral; since all features are not necessarily "qualities") of a self-organizing system emerge and have certain effects, i.e. a complex reflective relationships is established between the trigger of self-organization (the reflected), the emergent features (the result of reflection) and the function the new features fulfill for the system in its adaptation to its environment. We have defined this relationship as information, self-organizing systems are information-producing systems, information is not a pre-existing, stabile property of a complex system;
- *fault tolerance*: outside a critical phase, the structure of the system is relatively stable concerning local disturbances and a change of boundary conditions;
- *openness*: self-organization can only take place if the system imports entropy which is transformed, as a result energy is exported or as Prigogine says dissipated;
- *symmetry breaking*: the emerging structures have less symmetry than the foundational laws of the system;
- *inner conditionality*: self-organizing systems are influenced by their inner conditions and the boundary conditions from their environment;
- *relative chance*: there is a dialectic of chance and necessity in self-organizing systems; certain aspects are determined, whereas others are relatively open and according to chance;
- *complexity*: the complexity of a system depends on the number of its elements and connections between the elements (the system´s structure). There are three levels of complexity: 1. there is self-organisation and emergence in complex systems, 2. complex systems are not organized centrally, but in a distributed manner; there are many connections between the system´s parts, 3. it is difficult to model complex systems and to predict their behavior even if one knows to a large extent the parts of such systems and the connections between the parts.

REFERENCES

American Heritage Dictionary: emerge. Available from: http://www.bartleby.com/61/8/I0070800.html

Arshinov, V. and Fuchs, C. Causality, Emergence, Self-Organization. Available from: http://www.self-organization.org/results/book/EmergenceCausality Self-Organisation.pdf

Business Dictionary: vertical integration. http://www.businessdictionary.com/definition/vertical-integration.html

Special Interest Group on Anticipation of BISC. Definitions of Anticipation. http://www.anticipation.info/

Bishop, M. A view inside the Chinese room. http://www.doc.gold.ac.uk/~mas02mb/Selected%20Papers/2004%20Short%20CRA.pdf

Bonabeau, E., Dessalles, J.-L., and Grumbach, A. (1995). Characterising Emergent Phenomena (1) *A Critical Review. Revue Internationale de Systémique*, 9 (3), pp. 327-346. http://perso.telecomparistech.fr/~jld/papiers/pap.cogni/Dessalles_95111402.pdf

Brunner, K.A. and Klauninger, B. *An Integrative Image of Causality and Emergence*. In: Arshinov, V. and

C. Fuchs (eds.) Causality, Emergence, Self-Organization.http://www.selforganization.org/results/book/EmergenceCausalitySelf-Organisation.pdf

Buchanan, M. (2007a). The Social Atom: Why the Rich Get Richer, Cheaters Get Caught, and Your Neighbor Usually Looks Like You. Bloomsbury.

Buchanan, M. (2007b). The Social Atom (blog) http://thesocialatom.blogspot.com/2007/02/key-ideas.html

Copeland, B. J. (2008). Computation. Chapter 1 (pp. 1-17) of The Blackwell Guide to the Philosophy of Computing and Information. L. Floridi (ed.) http://www.blackwellpublishing.com/pci/downloads/SampleChapter.pdf

Deguet J., Demazeau, Y. and Magnin, L. (2006). Elements about the Emergence Issue: A Survey of Emergence Definitions. Complexus 2006;3:24–31. http://content.karger.com/ProdukteDB/produkte.asp?Aktion=ShowFulltext&ProduktNr=227088&Ausgabe=232060&ArtikelNr=94185 I O

Dessalles, J-L., Ferber, J. and Phan, D. (2008). Emergence in agent based computational social science: conceptual, formal and diagrammatic analysis. In: Y. Shyan & A. Yang (Eds), *Intelligent complex adaptive systems*. IGI Global, 255-299.

Feld, S.L. Looking at Patterns, Not People. American Scientist.http://www.americanscientist.org/bookshelf/pub/looking-at-patterns-not-people S

Fenzl, N. Emergence and Self-Organisation of Complex Systems. The Role of Energy Flows and Information. In: Arshinov, V. and C. Fuchs (eds.) Causality, Emergence, Self-Organization. http://www.selforganization.org/results/book/EmergenceCausalitySelf-Organisation.pdf

Fuchs, C. Dialectical Philosophy and Self Organization. In: Arshinov, V. and C. Fuchs (eds.) Causality, Emergence, Self-Organization. http://www.self-organization.org/results/book/EmergenceCausalitySelf-Organisation.pdf

Ghasem-Aghaee, N. and T.I. Ören, (2003). Towards Fuzzy Agents with Dynamic Personality for Human Behavior Simulation. *Proceedings of the 2003 Summer Computer Simulation Conference*, Montreal, PQ, Canada, July 20-24, 2003, 3-10. http://www.site.uottawa.ca/~oren/pubs/pubs-2003-02-SCSC-fuzzy-agents.pdf

Goldstein, J. (1999). Emergence as a Construct: History and Issues. Emergence, 1(1), pp. 49-72. http://www.questia.com/PM.qst?a=o&d=77010539

Jin, S., Huang, H., Fan, G. (2008). Emergence-Oriented Research on Multi-Agent Systems and Its State of Arts, Chinese Journal of Computers, 6, pp. 881—895).

Knuth, D. (1968). The Art of Computer Programming, vol. 1 – Fundamental Algorithms, Addison-Wesley, Reading, MA.

Lim, A. (2007). Decision Support Under Uncertainty Using Exploratory Multisimulation with Multiresolution Multistage Models. AFRL-IF-RS-TR-2007-198. Final Technical Report. http://stinet.dtic.mil/cgibin/GetTRDoc?AD=ADA473361&Location=U2&doc=GetTRDoc.pdf

McLeod, J. (1968). Simulation – The Dynamic Modeling of Ideas and Systems with Computers.McGraw Hill, new York, NY.

(MW-emerge) Merriam-Webster. http://www.merriam-webster.com/cgi.bin/dictionary?book=Dictionary&va=emerge

(OD-emerge) Compact Oxford Dictionary. http://www.askoxford.com/concise_oed/emerge?view=uk

Ören, T.I. (2001 – Invited contribution). Towards a Modelling Formalism for Conflict Management. In: Discrete Event Modeling and Simulation: A Tapestry of Systems and AI-based Theories and Methodologies. H.S. Sarjoughian and F.E. Cellier (eds.), Springer-Verlag, New York, pp. 93-106.

Ören, T.I. (2005 – Invited Keynote Article). Maturing Phase of the Modeling and Simulation Discipline. In: Proceedings of: ASC - Asian Simulation Conference 2005 (The Sixth International Conference on System Simulation and Scientific Computing (ICSC'2005), 2005 October 24-27, Beijing, P.R. China, International Academic Publishers - World Publishing Corporation, Beijing, P.R. China, pp. 72-85.

Ören, T.I. (2006). Body of Knowledge of Modeling and Simulation (M&SBOK): Pragmatic Aspects. *Proc. EMSS 2006 - 2nd European Modeling and Simulation Symposium*, pp. 327-336. October 4-6, Barcelona, Spain.

Ören, T.I. (2009 Jan.-In Press). Uses of Simulation. In: Principles of Modeling and Simulation: A Multidisciplinary Approach, by. John A. Sokolowski and Catherine M. Banks (eds.) (All Chapters by Invited Contributors). John Wiley and Sons, Inc. New Jersey.

Ören, T.I.; Yilmaz, L. (2004). Behavioral Anticipation in Agent Simulation. *Proceedings of the 2004 Winter Simulation Conference*, Washington, D.C., December 5-8, 2004, pp. 801-806. http://www.site.uottawa.ca/~oren/pubs/2004/07-WSC-anticip.pdf

Rey, A. (2005). Le destin d'un métaphore. Special Issue of Sciences et Avenir: L'énigme de l'émergence, No. 143, July/August 2005, p. 81.

Rosen, R. (1985). Anticipatory Systems. Pergamon Press.

(S&A, 2005) Special Issue of Sciences et Avenir: L'énigme de l'émergence, No. 143, July/August 2005.

Sawyer, K. (2004). The Mechanisms of Emergence. Philosophy of the Social Sciences, 34 (2), pp.260-282.http://artsci.wustl.edu/~ksawyer/PDFs/mechanisms.pdf

(SEP-emergent) http://www.science.uva.nl/~seop/entries/properties

-emergent/ (First published Sep 24, 2002; substantive revision Oct 23, 2006)

Yilmaz, L., A. Lim, S. Bowen, and T.I. Ören (2007). Requirements and Design Principles for Multisimulation with Multiresolution, Multistage Multimodels Proceedings of the 2007 Winter Simulation Conference, pp. 823-832, December 9-12, Washington, D.C.

Yilmaz, L. and Ören, T.I. (2004). Exploring Agent-Supported Simulation Brokering on the Semantic Web: Foundations for a Dynamic Composability Approach, WSC 2004 - Winter Simulation Conference, Washington, D.C., December 5-8, 2004, pp. 766-773.

Yilmaz, L., Ören, T.I. and Ghasem-Aghaee, N. (2006). Sim-ulation-Based Problem Solving Environments for Conflict Studies: Toward Exploratory Multisimulation with Dynamic Simulation Updating. *Simulation, and Gaming Journal.*37 (4), pp.534-556.

Yilmaz, L., Tolk, A. (2007). A Unifying Multimodel Taxonomy and Agent-Supported Multisimulation Strategy for Decision-Support. Intelligent Decision Making: An AI-Based Approach 2008: Springer-Verlag, pp.193-226.

AUTHORS BIOGRAPHIES

TUNCER ÖREN is a professor emeritus of Computer Science at the University of Ottawa. His current **research** activities include (1) advanced M&S methodologies such as: multimodels (to encapsulate several aspects of models), multisimulation (to allow simultaneous simulation of several aspects of systems), and emergence; (2) agent-directed simulation; (3) cognitive simulation (including simulation of human behavior by fuzzy agents, agents with dynamic personality and emotions, agents with perception, anticipation, and understanding abilities); and (4) reliability and quality assurance in M&S and user/system interfaces. He has **also contributed** in Ethics in simulation as the lead author of the Code of Professional Ethics for Simulationists, M&S Body of Knowledge, and multilingual M&S dictionaries. He is the founding director of the M&SNet of SCS. He has over 350 **publications** (some translated in Chinese, German and Turkish) and has been active in over 370 **conferences** and seminars held in 30 countries. He received "Information Age Award" from the Turkish Ministry of Culture (1991), Distinguished Service Award from SCS (2006) and plaques and certificates of appreciation from organizations including ACM, AECL, AFCEA, and NATO; and is recognized by IBM Canada as a Pioneer of Computing in Canada (2005). His homepage is: http://www.site.uottawa.ca/~oren/.

FRANCESCO LONGO took the degree in Mechanical Engineering from University of Calabria (2002) and the PhD in Industrial Engineering (2005). He is currently researcher at the Mechanical Department (Industrial Engineering Section) of University of Calabria. His research interests regard modeling & simulation of manufacturing systems and supply chain management, vulnerability and resilience, DOE, ANOVA. He is Responsible of the Modeling & Simulation Center – Laboratory of Enterprise Solutions (MSC-LES), member organization of the MS&Net (McLeod Modeling & Simulation Network) He is also member of the Society for Computer Simulation International and Liophant Simulation.

Performance Evaluation of Quantum Cascaded Lasers through VisSim Modeling

Mohamed B. El_Mashade(a), Imbaby I. Mahamoud(b), & Mohamed S. El_Tokhy(b)

(a) Electrical Engineering Dept., Al Azhar University, Nasr City, Cairo, Egypt
(b) Engineering Dept. NRC, EAEA, Inchas, Egypt

engtokhy@Gmail.com

ABSTRACT

Our goal in this paper is to evaluate the performance of quantum cascaded lasers (QCL's). The tools that we are used are the VisSim technique along with the block diagram programming procedures. The benefits of using this modeling language are the simplicity of carrying out the performance's measurement through computer simulation instead of setting up a practical procedure which becomes expensive as well as the difficulty of its management. The implemented models can help designers and scientists to optimize their devices to meet their requirements.

Keywords: *Quantum Cascaded Lasers (QCLs), Modeling, Block Diagram Programming, Intersubband transitions.*

1. INTRODUCTION

Quantum Cascaded lasers (QCLs) are new light sources based on intersubband transitions in quantum wells (QWs) [1]. These devices emit at wavelengths covering most of the mid-infrared (MIR) and part of the far-infrared (FIR) electromagnetic spectrum [2]. The extension to the far-infrared range is of particular interest due to the lack of narrow-band, powerful, and compact sources in this wavelength range. However, it presents a formidable challenge related to physics issues and technical difficulties [3].

The advent of the quantum cascade lasers (QCLs) with emission wavelengths available in the infrared range from 3 μm through more than 100 μm opens up the possibility of exploiting infrared atmospheric transparency windows for free space optical communications [4-5], trace gas analysis for pollution monitoring, environmental sensing, medical diagnostics, automobile applications military applications and wireless optical communications [6].

Additionally, the trend in the world was directed towards the quantum cascaded lasers because they have a several advantages over conventional laser diode, such as their high speed digital modulation and results on MIR optical wireless communication links, which demonstrate the possibility of reliably transmitting complex multimedia data streams [7].

Experimental Setups for measuring the characteristics of QCLs are reported [8]. However it is shown difficulties in tuning and cost much. In this work block diagram technique is used to overcome the above mentioned complexity. Models are designed for the transport process, and carrier densities for QCLs devices. In this paper, the tunneling transition and population inversion are discussed in details. We focused here on improving the characteristics of QCLs, by exploiting the parameters that have a large effect on the performance of QCLs through our developed models. In this paper, the operational principles of quantum cascaded lasers were presented. To build a self-sustained oscillator like a laser, so, it is important that the condition of population inversion is satisfied. Block diagram models are implemented by ViSsim describing the lasing characteristics of QCLs. This paper is organized as follows: In section *2,* we present the basic assumptions and model description. Proposed Simulator for Quantum Cascaded Laser including Tunneling process within QCLs model, and Population inversion model are considered in section *3.* We summarize our results in section *4.* Section *5* is devoted for conclusion.

2. BASIC ASSUMPTIONS AND MODEL DESCRIPTION

QCLs under consideration are complex devices, whose core is a multiple quantum wells (MQWs) [9]. Basically, QCLs containing a series of repeated InGaAs wells sandwiched between much thicker layers of the alloy semiconductor InAlAs barriers. But research has been expanded to other material systems, ultra-high-speed operation, and the exploration of different frequency ranges [10].

We assume that each quantum well (QW) contains three subband levels. The transitions between these subband levels are at equilibrium in the case of no bias and reach the flat condition, when the correct bias is applied [11]. It is assumed that the thermal effect of mobile carriers and thermally populated injector states of energy width are neglected. For the purpose of comparison, the QCLs parameters are chosen in such a way that they are compatible with the experimental ones.

As [10], the device realized by InGaAs/InAlAs lattice matched to InP, is constituted by one basic structure, which is known as a period, and this period is repeated several times. Each period is composed of two regions:

active region and relaxation-injection region. In the active region, the optical transition occurs and in the other one, the carriers can relax after having completed the optical transition. Since the injector region is highly doped, it can be acting as an electron reservoir and in turn it reinjects the electrons to the next period. This construction allows the electron to be recycling.

Additionally, the injected carriers from the injector region; owing to resonant tunneling, to the upper lasing state of the active region, state $|3\rangle$, they can relax to state $|2\rangle$ by means of photon-assisted tunneling or by scattering. This scattering is mainly due to longitudinal (LO) phonon, given that $E_{32} \geq \hbar\omega_{LO}$ is satisfied.

In the proposed model, the diagonal transitions are described by using three quantum wells. The Injector region contains five symmetrical QWs. The excitation energy between states 2 and 1 in each QW is assumed to be comparable with the phonon energy (~36meV). The applied biasing voltage should be sufficient to cause a tunneling into QCLs. In most cases of QCLs, the characteristic temperature "T_0" depends on the chosen compound material. QCLs has the advantage of a large T_0 that allows it to operate at higher temperature [12].

3. PROPOSED MODELS FOR QUANTUM CASCADED LASERS

Proposed programs for modeling and simulation of QCLs behavior is partially implemented in VisSim environment. VisSim is a visual block diagram for nonlinear dynamic simulation. The basic part of this diagram is based on what is known as a block. This block allows users to create their corresponding one in C/C++. In this environment, the system is modeled by the graphical interconnection of function blocks. For flexibility, variables are used to denote system parameters and then are assigned values in a separate compound blocks. Once the underlined problem was represented by its group of blocks, it is ready to be evaluated through the VisSim which is internally programmed. The program can be distributed with VisSim viewer or through generated C code from VisSim block diagram, which means that it doesn't depend on the VisSim environment [13].

3.1. Tunneling Process within QCLs Model

Block diagram modeling technique is used to represent the basic equation of tunnelling current derived by Kazarinov and Suris. The developed model is used to study the effect of current on both tunneling rate and splitting energy. This allows detection of the minimal biasing current which produces tunnelling while in the same time reduces such splitting energy.

3.1.1. Resonant Tunneling

The resonant tunneling of the electrons between the ground state of the injector and the upper lasing state of the active region was derived by Kazarinov and suris [14] is expressed as follows:

$$J_{\max} = eN_s \frac{2\Omega_{13}^2 \tau_p}{1 + \Delta^2 \tau_p^2 + 4|\Omega_{13}|^2 \tau_p \tau_3} \qquad (1)$$

$$\hbar\Delta = E_1(F) - E_{33}(F) = qd(F - F_r) \qquad (2)$$

$$d = |Z_{gg} - Z_{33}| \qquad (3)$$

here J_{max}, e, N_s, τ_3, τ_p, $2\Omega_{i3}^2 \tau_p$, $\hbar\Delta$, d, F, and F_r denotes the maximum injected current, the electron charge, the sheet carrier density, the upper state life time, the in-plane dephasing time , the tunneling rate, the energy detuning from resonance, the spatial separation between the cenroids of the two wavefunctions, the average electric field applied over the distance d and the electric field which brings the upper lasing state 3 and the ground state g of the injector into resonance respectively. Block diagram model describing the maximum current with the tunneling transport is depicted in Fig.(1).

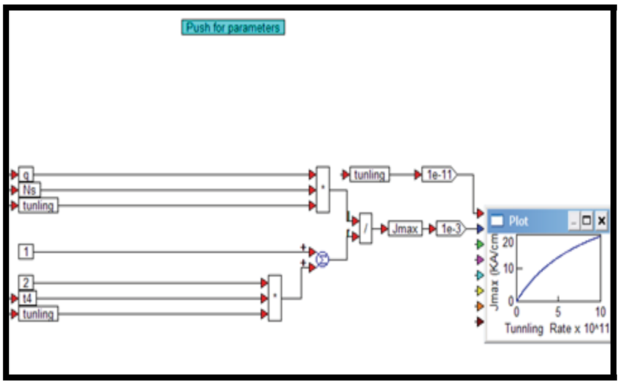


Figure 1: Model of Tunneling Rate and Maximum Current Density.

3.1.2. Splitting Energy and coupling parameters

The splitting energy ($2\ \hbar\Omega_{i3}$), between the two concerned states (ground state of the injector and the upper lasing state of the active region) and the J_{max}, is described by the following relation that derived from Eq.(1) after simple calculation.

$$J_{\max} = \frac{q\ Ns\ S^2 \tau_p}{\dfrac{h^2}{2\pi^2} + 2S^2 \tau_p \tau_3} \qquad (4)$$

$S = 2\ \hbar\Omega_{i3}$ is the splitting energy. Block diagram model describing the maximum current with the splitting energy is depicted in Fig. (2).

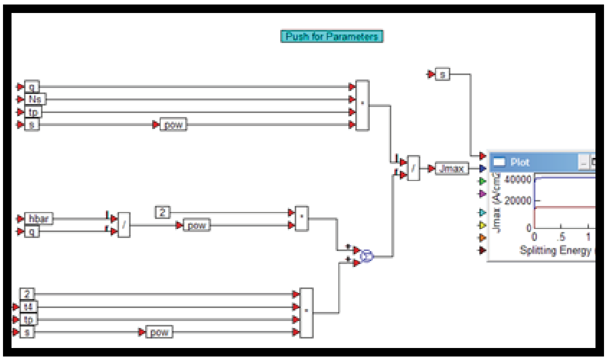


Figure 2: Block Diagram Model Of Splitting Energy And Maximum Current Density.

3.2. Population inversion model

A model for the bias current and carriers densities in quantum cascaded laser is presented. The non-linear rate equations for the electron densities, as a function of photon density, can be written as [8].

$$\frac{dn_3}{dt}=\frac{J_{in}}{q}-\frac{n_3}{\tau_{32}}-\frac{n_3}{\tau_{31}}-\Gamma\upsilon_g\ a\ (n_3-n_2)(S_p+\frac{n_{sp}}{WL}) \qquad (5)$$

$$\frac{dn_2}{dt}=\frac{n_3}{\tau_{32}}-\frac{n_2}{\tau_{21}}+\Gamma\upsilon_g\ a\ (n_3-n_2)(S_p+\frac{n_{sp}}{WL}) \qquad (6)$$

In the above expression, L, v_g, a, Γ, n_{sp}, n_3 & n_2, S_p, τ_{32}, $\tau_{31,}$ and $\tau_{21,}$ denotes the length of the cavity, group velocity of the lasing modes, the differential gain (cm) contributed by a single gain stage, the mode confinement factor for the gain stage, the spontaneous emission factor, are the carrier densities in subbands 3 and 2, the photon density per unit width inside the optical cavity, the relaxation time of the optical transition of the electron between the states (3 and 2), the life time of the optical transition of the electron between the states (3 and 1) and the life time of the optical transition of the electron between the states (2 and 1), respectively. Block diagram model describes the effect of population inversion in QCLs system and its inherent operation are shown in Fig.(3).

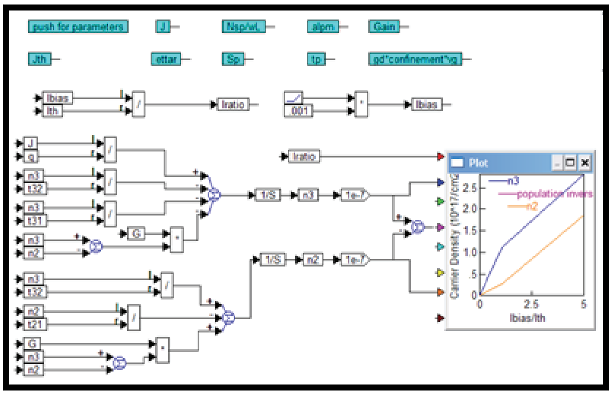


Figure 3: The Basic Model Describing the Rate Equations of Qcls.

4. SIMULATION RESULTS AND DISCUSSION

In this section, the results obtained from our developed models are presented. Our starting point of view is the relation between the tunneling rate and J_{max}, at different doping densities. Fig (4) depicts the plotting of the biasing current and the tunneling rate as a function of doping levels. It is noted that the alignment between the injector and the active regions improves as the current increases. This means that the splitting energy between the upper state of the active region and the ground state in the injector region decreases. As a result of this, a channel for the electrons will be created between the two concerned states and consequently the tunneling rate becomes higher. As a conclusion from the results of this figure, we can enhance the tunneling rate by changing some of other important parameters such as doping. As the doping increases, the required maximum current increases. This in turn will lead for the tunneling rate to become higher than before.

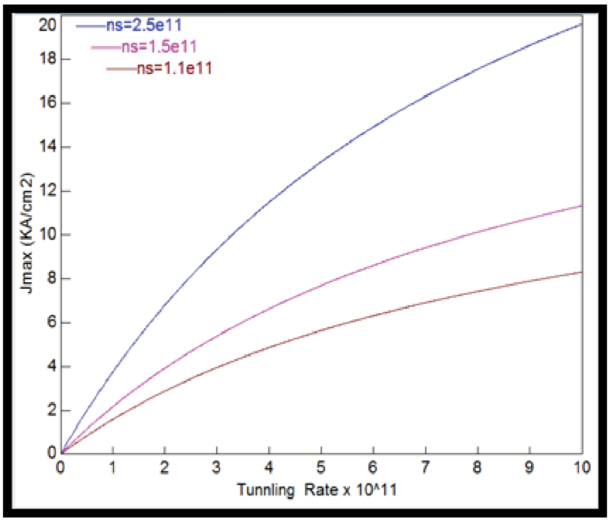


Figure 4: Variation of the Maximum Injected Current as a Function of the Tunneling Rate at Different Doping Levels.

Splitting energy (2 Ω_{ij} ħ) between the two concerned states (ground state of the injector and the upper lasing state of the active region), as a function of the maximum current density is depicted in Fig. (5). Since the splitting energy of the concerned states is increased, a large value of the current will be needed to compensate for the difference in energy between these two states. At the same time, two regions can be identified in the curve: one of them, which is characterized by the smallest splitting energy, the injection is very efficient and J_{max} is increased to balance the difference in energy between the two concerned states. In the second region, if the splitting energy increases above a certain limit, the injection current has no effect on it. This means that the injected current is not sufficient to make the two states near to one another. To overcome this difficulty, the doping effect must be used. As the doping increases, the carriers that transport to the active region will be increased. Consequently, higher performance of the device is obtained. It is of importance to note that as the coupling becomes stronger, the fastest injection of carriers to the upper state of the laser transition can be achieved and this in turn makes the lasing action more flexible.

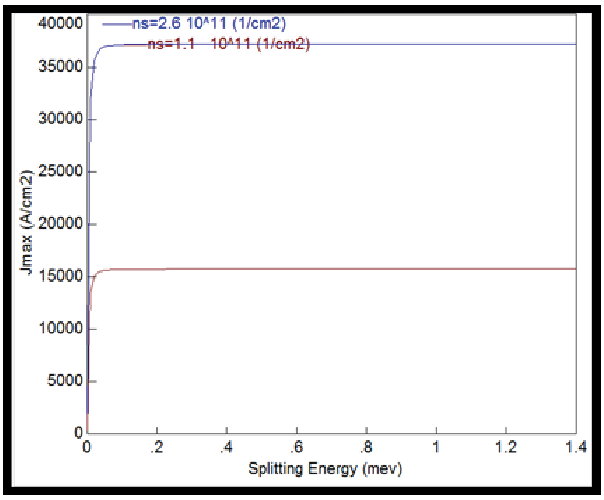


Figure 5: Splitting Energy As a Function of the Maximum Injected Current Parametric In Doping Density.

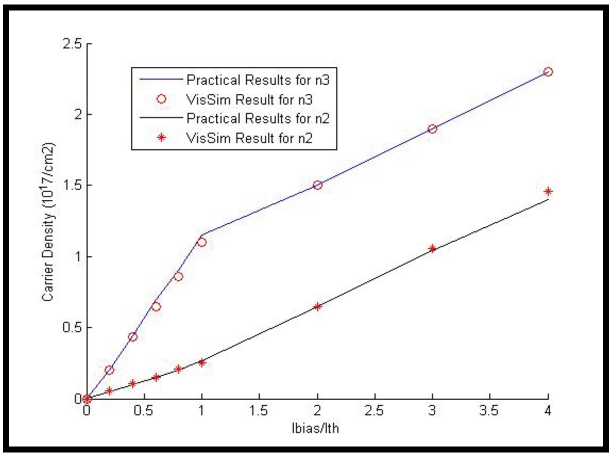


Figure 6: Variation of the Carrier Densities Along With the Population Inversion as a Function of the Bias Current.

We noticed that to enhance the capabilities of the system, doping of the injector region is necessary to maintain a steady-state population inversion as dictated by charge neutrality. And the level of doping must be carefully optimized to give minimal absorption and leakage.

The rate of change of electron densities in levels 3 and 2 with the bias current is shown in Fig.(6). From this figure, as the electrical current increases, the number of electrons in the upper subband increases and this in turn leads to enhance the population inversion. As a consequence of this, the number of photons will be raised and the lasing characteristics will be improved. The results of the underlined figure exhibit a bending at threshold. After threshold the carrier densities will be further increased with current. For this reason each curve contains two lines with different slopes. These bendings in the rate of increasing of electron densities, with the bias current, result in discontinuities in the values of the differential resistance of the laser at threshold. Additionally, this behavior of theoretical results is in a good agreement with that published in [8].

At resonance, all electrons between the injector ground state and the upper laser state are equally distributed**.** While below resonance, the carriers reside in the upper laser state 3 is small compared to that state above threshold as Fig. (7) demonstrates. Below threshold, the carriers in subband n_2 are larger than that above threshold. This is because the carriers above threshold don't attain in this state. The energy of the electron is made equal to the phonon energy (the main scattering mechanism). So, there is a fast decay in this state from electrons as depicted in Fig.(7). Even though, the gain is clamped at its value which it attains at threshold. In the same time, the electron densities are continuously increasing with the bias current. As a result, an increase in the injected current density in QCLs does not only lead to an increase in the photon emission rate but it also leads to an increase in the rate of non-radiative transitions. Therefore, QCLs tend to have radiative efficiencies η_r significantly smaller than unity.

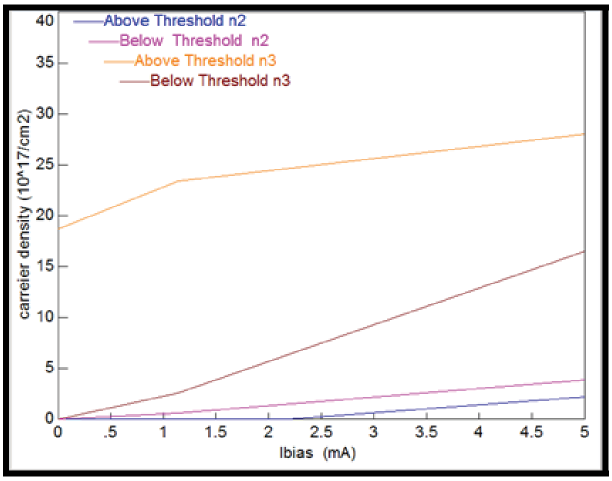


Figure 7: Dependent of the Carrier Densities On The Lasing State Above And Below Threshold.

Also, Figs.(7) shows that the population is much more widely distributed over the different levels below threshold. Also, above threshold, the carrier density in the quantum wells is strongly damped, and only the carrier density in the separate confinement heterostructure region provides negative feedback to suppress the noise associated with carrier injection.

5. CONCLUSION

In this work, models describing the performance of QCLs are developed. Block diagram programming technique is employed to implement these models. To ensure the validity of the model, comparison with the published practical result is performed. The models are used to study the effect of device parameters on performance characteristics of the QCLs within user friendly graphical environment. Results show the effectiveness of methodology introduced. As an example, doping effect which received little attention in the literature is investigated.

REFERENCES

L. A. Dunbar, V. Moreau, R. Ferrini and R. Houdr´e, 2005. Design, fabrication and optical characterization of quantum cascade lasers at terahertz frequencies using photonic crystal reflectors. *Optics Express*, 13(22), October.

M. Razeghi, and S. Slivken, 2003. High powers quantum cascaded lasers grown by Gas MBE. *Opto-Electronics Review,* 11(2), pp. 85–91.

R. Colombelli, F. Capasso, C. Gmachl, A. L. Hutchinson, D. L. Sivco, A. Tredicucci, M. C. Wanke, A. M. Sergent, and A. Y. Cho, 2001. Far-infrared surface-plasmon quantum-cascade lasers at 21.5 μm and 24 μm wavelengths. *Appl. Phys. Lett.*, 78(18), April .

R. Martini and E.A. Whittaker, 2005. Quantum cascade laser-based free space optical mmunications. *Journal of Optical and Fiber Communications Reports*, 2(4), October.

S. Blaser, D. Hofstetter, M. Beck and J. Faist, 2001. Free-space optical data link using Peltiercooled quantum cascade laser. *Electronics Letters*, 37(12), June.

M. S. Taubman, T. L. Myers, B. D. Cannon, and R. M. Williams, F. Capasso, C. Gmachl, D. L. Sivco, and A. Y. Cho, 2002. Frequency stabilization of quantum-cascade lasers by use of optical cavities. *Optics Letters*, 27(24), December.

F. Capasso, R. Paiella, R. Martini, R. Colombelli, C. Gmachl, T. L. Myers, M. S. Taubman, R. M.Williams, C. G. Bethea, K. Unterrainer, H. Y. Hwang, D. L. Sivco, A. Y. Cho, A. M. Sergent, H. C. Liu, and E.. A. Whittaker, 2002. Quantum Cascade Lasers: Ultrahigh-Speed Operation, Optical Wireless Communication, Narrow Linewidth, and Far-Infrared Emission. *IEEE Journal of Quantum Electronics*, 38(6), June.

Farhan Rana, Rajeev J. Ram, 2007. Theory of Current Noise and Photon Noise in Quantum Cascade Lasers. *arXiv:cond-mat/0108236v1 [cond-mat.mes-hall]*, July.

Rita Claudia Iotti, and Fausto Rossi, 2001. Nature of Charge Transport in Quantum-Cascade Lasers. *Phys. Rev. Lett*, 87(14), October.

F. Eickemeyer, 2005. *Ultrafast dynamics of coherent intersubband polarizations in quantum wells and quantum cascade laser structures*. Thesis (PhD). Neuchatel university.

Markfox, 2001. Optical properties of solids. Oxford University, press, book, ch. 6.

J.Faist, F. Capasso, Carlo Sirtori, Deborah L. Sivco, Albert L. Hutchinson, and Alfred Y. Cho, 1995. Vertical transition quantum cascade laser with Bragg confined excited state. *Appl. Phys. Lett.*, 66(5), January.

H.M. Abdull-kader, A.B.El-Sisi, K.A.Mostafa, I I.Mahmoud, 2007. Implementation of A real-time Simulator for dynamic systems. *IJCI*, 1(1), July.

T. Aellen, M. Beck, N. Hoyler, M. Giovannini, and J. Faistb, 2006. Doping in quantum cascade lasers I. InAlAs–InGaAs/InP midinfrared devices. *Journal of Applied Physics*.

SELECTING THE OPTIMUM BY SEARCHING AND RANKING PROCEDURES IN SIMULATION-BASED OPTIMIZATION

Pasquale Legato [(a)] **and Rina Mary Mazza** [(b)]

[(a)] DEIS, Università della Calabria, Via P. Bucci 41C, 87036, Rende (CS), Italy
[(b)] DEIS, Università della Calabria, Via P. Bucci 41C, 87036, Rende (CS), Italy

[(a)]legato@deis.unical.it, [(b)]rmazza@deis.unical.it

ABSTRACT

Given a set of competing system alternatives to be evaluated and compared via simulation, *Ranking and Selection* (R&S) procedures are commonly applied to select the best system with respect to a predefined performance measure. In this paper we focus on two major classes of R&S techniques usually referred to as the *subset selection* and *indifference-zone* formulations. In particular, we discuss the performance of primitive and combined procedures that, at every iteration, evaluate different system configurations by sampling multiple or single additional simulation output observations to deal with complex systems. Procedure application is presented for different test cases in which either a small number of system configurations are known *a priori* or a large number of configurations are actually generated during simulation run by means of simulation-based optimization algorithms. Preliminary numerical results are given with reference to performance measures within the sub-systems of a real-world complex logistic system.

Keywords: discrete-event simulation, combinatorial optimization, statistics, metaheuristics.

1. INTRODUCTION

In a wide variety of application fields, such as logistics, production engineering and plant layout, computer-communication systems organization and others, discrete-event simulation is well recognized as an effective planning and control tool. By commonly adopting a simple *what-if* approach, simulation is also used for supporting decisions in system management and performance evaluation of policies and rules for resource allocation. The main reason for such versatility of simulation lies in its capability of representing the process of interest in a realistic, dynamic framework subjected to several elements of uncertainty and randomly occurring events and activity durations.

On the other hand, operations research models based on integer programming formulations (Nehmauser and Wolsey 1988) of combinatorial optimization problems are well consolidated as stand-alone tools for supporting specific decisions of resource allocation and activity scheduling. This is especially successful whenever the idea of fixing the process of interest at any given instant, in which the modeller may identify a set of resources and a set of tasks in a static, deterministic operational framework, results as a cost-effective choice.

The ultimate effort to spark and steer strong interplay and even overlap between these neighbouring fields of operations research is known as simulation-based optimization (Andradóttir 1998). This methodology consists in optimizing an expected performance measure based on outputs from stochastic simulations of any given system/process, whose dynamic behaviour is partially defined by some decision variables and constraints that could be optimally determined by an IP model. To fix ideas, one may think of a queuing network model of a general job shop system where the performance measure is the expected value of the *makespan* or the system *throughput* and should be estimated by solving the queuing network model proposed via simulation (Laganà, Legato, Pisacane et al. 2006; Canonaco, Legato and Mazza 2007; Canonaco, Legato, Mazza et al. 2008). Assuming that the allocation of servers to different queuing stations, as well as the selection of the service discipline are, respectively, the major resources to be allocated and the major scheduling policies to be organized, then it should not be difficult to recognize that a possible formulation of the simulation-based optimization problem would require replacing the objective function of the formulated combinatorial optimization problem with the following: $min\, E[f(\theta)]$. The "f" function also accounts for implicit additional process features and queuing phenomena when searching for the optimal vector of decision variables, θ (representing resource allocation and scheduling policies).

Practically speaking, simulation-based optimization methods feature a "comparison step" between alternative feasible solutions and policies which is always based on the use of statistics to estimate the expected performance measure of interest (throughput, makespan, and so on). The above statistics are computed on a certain number of observations. Since these observations are random variates returned from a simulation process, at each comparison there are no guarantees of selecting the true best (optimal) solution, despite it being truly representative of the best system configuration. Therefore, at the comparison step of a simulation-based optimization algorithm one should carefully design a statistical procedure to perform a correct selection with at least a user-specified probability. To this end, we discuss some *Ranking and Selection* (R&S) procedures (Goldsman, Kim, Marshall et al. 2002) which are commonly applied to select the system with the "best" (i.e. greatest or smallest) expected value of the predefined performance index (Kim and Nelson 2003).

The paper is organized as follows. In Section 2 we discuss the two major R&S approaches when all alternative simulated system designs are known in advance. In Section 3 we propose the combination of R&S and simulation-based optimization procedures to deal with system designs that are revealed during simulation experiments. Numerical experiments are presented in Section 4 and conclusions are drawn in Section 5.

2. RANKING AND SELECTION

In literature, homogeneity tests (Milton and Arnold 1986) are conventionally applied to assess whether there are statistically significant differences in various populations with respect to some characteristics. However, they provide no information in the prospect of selecting the "best" of these populations. Bearing in mind this expectation, *Ranking and Selection* techniques are the next step to take when searching for a decision procedure that allows to perform a correct selection at a pre-assigned level of probability.

Most of the research work in R&S can be classified into the following general approaches:

- *subset selection* procedures, which aim at producing a subset of (small) random size that contains the best system, with a user-specified probability;
- *indifference-zone* procedures, where either the best or whatever solution evaluated within a fixed distance from the best can be selected, with a user-specified probability.

When operating a selection of the best system or a subset of the best among a set of simulated competing alternatives, using an R&S technique rather than another depends on which of the available procedures will most benefit a given objective or constraint set by the experimenter.

Whatever the objective, an "educated" choice of an R&S procedure also requires a good knowledge of the structure of the problem space in view of the fact that the said structure impacts on the performance of the procedures that can be used for problem solving.

Everything considered, the performance level of an R&S procedure is affected by:

- the probability of selecting the alternative which is truly representative of the best system configuration (PCS – probability of correct selection);
- the above probability returned within a given predetermined time budget;
- the existence of extreme configurations in which, for example, all solutions have an equal mean value or every solution is distant exactly delta units from the best (a.k.a. the *least favorable configuration*) or ordered solutions are equally spaced from one another;
- the difference between solutions which is assumed to be statistically insignificant;
- the structure of the problem space.

This stated, it is quite logical that different problems require different approaches. For example, in complex systems one of the following situations might occur: *i*) all the possible alternative system configurations are known before experimentation or *ii*) system configurations are revealed (meaning generated) during experimentation. Obviously, these cases also call for the use of specific (meaning different) procedures.

In this paper, we consider selecting the best system(s) according to a user-defined probability under a pre-assigned time budget, whenever the solutions are either all known *a priori* (see Bechhofer, Santner and Goldsman 1995 for a complete summary) or revealed during experimentation (Hong and Nelson 2007). For the former case, we examine two stand-alone R&S procedures that belong to the *subset-selection* and *indifference-zone* approaches; for the latter, we combine the above R&S procedures with a simulation-based optimization algorithm whose objective is to generate new alternative systems at run-time.

In both cases, as far as notation is concerned, we use k to call the number of alternative simulated system designs ($i = 1..k$); n the number of observations ($j = 1..n$) sampled from each system design; $\mu_1, \mu_2, ..., \mu_k$ the unknown k expected values of the performance measure of interest; $\mu_{[k]} \geq ... \geq \mu_{[1]}$ the ordered unknown k expected values of the performance measure of interest (i.e. the system design in position k is the greatest); $\overline{X}_k, ..., \overline{X}_1$ the sample means of the performance measure of interest for each system design; σ^2 the common unknown variance of the alternative system designs; $\overline{S}_k^2, ..., \overline{S}_1^2$ the sample variance of the performance measure of interest for each system design; $1-\alpha$ the confidence level (or user-specified probability P^*). In a maximization problem, we also use k to call the system with the best (meaning greatest) performance measure of interest.

It is worth observing that the basic underlying assumptions for all these R&S procedures, meaning independent and identically distributed normal data with common variance, usually depart from the realistic settings involved when simulating real-world systems. However, some important statistical results allow to extend the application of simulation-based optimization methods to problems in which simulation output data is not independent, nor normally distributed. These issues range from performing the proper process initialization (Law and Kelton 2000) to finding a consistent estimator for the sample variance (Meketon, and Schmeiser 1984; Goldsman, Meketon and Schruben 1990; Damerdji 1994; Song and Schmeiser 1995; Glynn and Whitt 1997; Steiger and Wilson 2002).

2.1. Subset-selection Procedures

Rather than claiming that one population is the best, perhaps it is more convenient to claim that one is confident that the best population is contained in a subset I of the $\{1,2,...,k\}$ competing simulated systems. Subset selection procedures are based on this logic. These R&S procedures aim at producing a subset of (small) random size that contains the best system, with a user-specified probability.

This R&S approach was first introduced by Gupta (1965) with the purpose of obtaining a subset $I \subseteq \{1,2,...,k\}$ according to which

$$P\{k \in I\} \geq 1-\alpha. \tag{1}$$

Basically, Gupta's idea was to include in I all the systems k that fall in the following interval:

$$\left[\bar{X}_k(n) - h\sigma\sqrt{\frac{2}{n}}, \quad \bar{X}_k(n)\right] \tag{2}$$

where σ is the common, known standard deviation and $\bar{X}_k(n)$ is the maximum among the sample means. Obviously, the most favorable case would be $|I| = 1$.

In order to guarantee (1), the value of h in (2) is determined as follows:

$$P\{k \in I\} =$$

$$= P\left\{\bar{X}_k(n) \geq \bar{X}_i(n) - h\sigma\sqrt{\frac{2}{n}}, \quad \forall i \neq k\right\} \tag{3}$$

$$= P\left\{\frac{\bar{X}_i(n) - \bar{X}_k(n) - (\mu_i - \mu_k)}{\sigma\sqrt{2/n}} \leq h - \frac{(\mu_i - \mu_k)}{\sigma\sqrt{2/n}}, \forall i \neq k\right\}. \tag{4}$$

If μ_k is the unknown performance measure of the "best" system, then $-\frac{(\mu_i - \mu_k)}{\sigma\sqrt{2/n}}$ is a positive value, thus

$$P\{k \in I\} \geq P\left\{\frac{\bar{X}_i(n) - \bar{X}_k(n) - (\mu_i - \mu_k)}{\sigma\sqrt{2/n}} \leq h, \forall i \neq k\right\} \tag{5}$$

and finally

$$P\{k \in I\} \geq P\{Z_i \leq h, \quad i = 1,2,...,k-1\} = 1-\alpha \tag{6}$$

where $(Z_1, Z_2, ..., Z_{k-1})$ are distributed according to a multivariate normal distribution with means equal to 0, variances equal to 1 and common pair-wise correlation equal to $1/2$. In order to guarantee (1), h must be the $1-\alpha$ quantile of the maximum value of $(Z_1, Z_2, ..., Z_{k-1})$.

The following pseudo-code provides a high-level description of Gupta's approach:

```
0. select $1-\alpha$, n and h;
1. take a random sample of n from each of the k systems;
2. compute an estimate of the performance index of interest for each of the k systems;
3. include a system in subset I if the system's sample mean falls in (2).
```

In the above procedure, the choice of $1-\alpha$ is left to the experimenter. Practically, $1-\alpha$ should be greater than or equal to 0.5, since any system could be included in I by simply tossing a fair coin. At the same time, $1-\alpha$ should also be greater than or equal to $1/k$ which is the probability of randomly selecting a system for inclusion in the subset. A pure empirical rule (Gibbons, Olkin and Sobel 1979) recommends $1-\alpha \geq 0.5 + (0.5/k)$.

2.2. Indifference-zone Procedures

Similar to any other selection procedure dealing with random variates returned from a simulation process, the indifference-zone based approach may or may not select the simulated system configuration which is truly representative of the best solution (if it does, then a correct selection (CS) is said to have been made). The novelty lies in the fact that this selection approach is statistically indifferent to which system configuration is chosen among a set of competing alternatives when these alternatives all fall within a fixed distance from the best solution.

This stated, let $P\{CS\}$ be the probability of correct selection and δ the indifference-zone chosen by the experimenter. In a maximization problem the probability of performing a correct selection with at least probability P^* is

$$P\{CS\} \hat{=} P\{\mu_k > \mu_i \forall i \neq k \mid \mu_k - \mu_i \geq \delta\} \geq P^*. \tag{7}$$

The probability of correct selection (7) was first computed in Rinott (1978) by resorting to numerical integration under the hypothesis of normality of the statistics involved. If $P(CS)$ is the probability that $\overline{X}_k(n_k)$ is the true “best” sample mean, namely it corresponds to $\overline{X}_{[k]}(n_k)$, then

$$P(CS) =$$

$$= P\left[\overline{X}_k(n_k) = \overline{X}_{[k]}(n_k)\right] \tag{8}$$

$$= P\left[\overline{X}_k(n_k) > \overline{X}_{k-1}(n_{k-1})\right], \text{ for short } P\left[\overline{X}_k > \overline{X}_{k-1}\right] \tag{9}$$

$$= P\left[\frac{\overline{X}_k - \mu_k}{\delta/h} > \frac{\overline{X}_{k-1} - \mu_{[k-1]}}{\delta/h} + \frac{\mu_{[k-1]} - \mu_{[k]}}{\delta/h}\right]. \tag{10}$$

Since $\left[\frac{\overline{X}_k - \mu_{[k]}}{\delta/h}\right] \mathrel{\hat{=}} T_k$ and $\left[\frac{\overline{X}_{k-1} - \mu_{[k-1]}}{\delta/h}\right] \mathrel{\hat{=}} T_{k-1}$ are distributed according to Student’s law with $n_k = n_{k-1} = \cdots = n_0$ degrees of freedom (Law and Kelton 2000) and since $\overline{X}_k$ and $\overline{X}_{k-1}$ are assumed to follow a Normal distribution, then

$$P(CS) =$$

$$= P\left[T_k - T_{k-1} > \frac{\mu_{[k-1]} - \mu_{[k]}}{\delta/h}\right] \tag{11}$$

$$= P\left[T_{k-1} < T_k + \frac{\mu_{[k]} - \mu_{[k-1]}}{\delta/h}\right]. \tag{12}$$

According to the total probability distribution conditioned on T_k,

$$P\left[T_{k-1} < T_k\right] =$$

$$= \int_{t=0}^{\infty} P\left[T_{k-1} \le t \mid t < T_k \le t + dt\right] * P\left[t < T_k \le t + dt\right] \tag{13}$$

$$= \int_{t=0}^{\infty} F_{T_{k-1}|T_k}(t) f_{T_k}^{(m)}(t)dt. \tag{14}$$

Because of independence between T_k and T_{k-1} then

$$= \int_{t=0}^{\infty} F_{T_{k-1}}(t) f_{T_k}(t)dt. \tag{15}$$

In the particular case (maximization) under examination

$$P\left[T_{k-1} < T_k + \frac{\mu_{[k]} - \mu_{[k-1]}}{\delta/h}\right] =$$

$$= \int_{t=0}^{\infty} F_{T_{k-1}}\left(t + \frac{\mu_{[k]} - \mu_{[k-1]}}{\delta/h}\right) f_{T_k}(t)dt. \tag{16}$$

Since $\mu_{[k]} - \mu_{[k-1]} \ge \delta$, the final result is

$$P(CS) \ge \int_{t=0}^{\infty} F_{T_{k-1}}(t + h) f_{T_k}(t)dt. \tag{17}$$

Note that equality is verified when $\mu_{[k]} - \mu_{[k-1]} = \delta$. If the integral is set equal to P^* and solved numerically for h, for different values of n (the number of observations taken from the system to compute the sample mean), the results can be tabled and read to obtain h, which is also known as Rinott’s constant. Numerical values for h are tabled in Wilcox (1984).

The following pseudo-code provides a high-level description of a two-stage indifference-zone procedure:

```
Stage 1
0.  select 1-$\alpha$,$\delta$, $n_0$ and h;
1.  take a random sample of $n_0$ from each system;
2.  compute the sample variance $S_i^2$ of the
    performance index of interest for each system;
3.  compute $N_i$=max($n_0$, $h^2S_i^2/\delta^2$);
Stage 2
4.  if $n_0$>=$\max_i N_i$ then select the system with the
    greatest sample mean otherwise take an
    additional sample of $N_i$-$n_0$ from each system i
    and then select the system with the greatest
    sample.
```

As shown at step 3, the total number of samples to be taken from each system mostly depends on sample variance and, thus, also on how the sample mean is computed. In two-stage R&S approaches, different methods are used for this purpose. Rinott (1978) uses a classic sample mean, while Dudewicz and Dalal (1975) use a weighted sample mean during the second stage. In (Canonaco, Legato and Mazza 2009), we investigate a moving-average sample mean whose early results are very promising.

3. SEQUENTIALLY REVEALED CONFIGURATIONS

The R&S procedures examined in the previous section are based on the common assumption that a (small) number of system configurations are known *a priori*. In this particular case, the guarantee of selecting the best or near-best alternative when all solutions have already been sampled and retained appears to be both very appealing and practicable. However, at times, a combinatorial, unknown number of configurations need to be explored. When this occurs, k different systems configurations (with $k \geq 1$) can be revealed sequentially during a simulation run by means of a so-called system generating algorithm (SGA). Under this hypothesis, should an exhaustive coverage of all the possible system combinations be not reasonable, nor affordable from a computational point of view, then metaheuristic-based approaches would have to be addressed. Examples of similar new, promising methodologies are *simulated annealing* (Ahmed and Alkhamis 2002; Andradóttir and Prudius 2005) and *adaptive balanced explorative and exploitative search* (Prudius and Andradóttir 2004; Prudius 2007). Although these algorithms are not used to perform an exhaustive coverage of the sample space, nor do they provide any sort of control running on which part of the feasible set is being explored, the solutions returned as final output are likely to belong to the set of optimal global solutions (Legato, Mazza and Trunfio 2008). This is a major issue in light of how demanding a statistical guarantee of correct selection in each iteration can be, especially when the number of candidate solutions visited by the optimization process is very large. As matter of fact, the new sequential selection procedures presented in (Hong and Nelson 2007) are applicable to small-scale optimization problems alone (with a number of systems ≤ 500), while extensions of these procedures to optimization problems with a very large number of alternatives is currently a subject of ongoing research efforts.

This stated, an overall scheme of the experimental framework, providing both system design generation and evaluation for selecting the best among these competing alternatives, can be summarized by the following pseudo-code.

```
0. set initial system design to be the best system design and set iteration equal to 0;
1. update iteration and generate k (k≥1) alternative system designs during the current iteration;
2. compare alternative system designs generated with current best system design and, eventually, update the best;
3. if stopping condition is met then exit, otherwise go to Step 1.
```

The R&S procedures examined in the previous section are used for system comparison in the above schema (Step 2), while the simulated annealing (SA) algorithm now described is used as SGA (Step 1).

```
0. set input parameters (initial Temperature, lower bound Temperature, time-budget) and best system design to be initial system design;
1. while elapsed time is less than time-budget and the Temperature is less than the lower bound value:
   (a) generate a system design from the current system design;
   (b) if the new design is “better”, then set best system design to be new design; else accept new design as best with probability p=exp(Δ/Temperature), where Δ is the difference between the two designs.
```

In brief, the SA approach is aimed to generate feasible schedules, explore them in a more or less restricted amount without getting caught in local minima and, finally, stop at a satisfactory solution.

4. NUMERICAL EXPERIMENTS

In this section, we present our efforts in searching for “intelligent” sample allocation when solving well-structured problems with significant constraints, especially within large, real-sized contexts. From a practical point of view, avoiding over-sampling affects the termination of the selection procedures and, thus, results are obtained with the least amount of simulation (i.e. execution time) possible.

Let us first examine the empirical performance of a primitive *Gupta-like* subset-selection procedure to select the best yard crane assignment and transfer policy in a terminal container. In particular, the objective is to select the policy which allows to minimize the maximum average time to complete stacking/retrieval operations of suitable batches of containers (BCT) in the yard. The scenario proposed features medium container traffic intensity and high crane transfer time among yard blocks. All experiments are carried out by setting $P^* = 0.90$, $k = 5$ and $n = 10$, under the realistic assumption of unknown, but common variance for each system design. To this end, Bartlett’s test (1937) has been used to verify the common variance assumption.

Table 1: Simulation Results for the Five Alternative Yard Policies

Policy *i*	N° of Observations			Average BCT (minutes)
	SSP	RP	OP	
Policy 1	10	31	10	97.369
Policy 2	10	27	17	91.043
Policy 3	**10**	**10**	**10**	**78.177**
Policy 4	10	32	26	100.052
Policy 5	10	48	17	92.343

According to the subset-selection procedure (SSP), the interval defined by (2) for this particular problem is $[78.117,\ 90.155]$, thus $I = \{3\}$. This result has also been compared with those returned by another two R&S procedures: Rinott’s procedure (RP) (1978) and our procedure (OP) (Canonaco, Legato and Mazza 2009). As one may easily calculate from Table 1, a cumulative (over all policies) number of 148 and 80 observations are required by these two procedures, respectively, whereas the SSP accomplishes the same result with only 50 observations. The case study just presented is representative of a typical situation where system configurations are well-spaced from each other, with respect to the performance metric adopted for comparison. Here one may recognize that the SSP allows to screen suitable configurations with a very limited number of observations.

We now present a second case which requires the combination of multi-stage R&S procedures with simulation-based optimization approaches since the alternative systems are revealed during the execution of the experiment.

In the combined procedure (CP) proposed herein, we pursue the idea of "efficient" sampling by basing the number of observations to be taken from each system on the convergence of process variance and whether this occurs within a certain number of simulation runs. In other words, at each iteration, this variance-weighted decisional mechanism decides to switch between adding a single (Chen and Kelton 2005) or multiple additional simulation output observations (Kim and Nelson 2003). With this multi-stage approach, the procedure is expected to terminate faster.

We use the above framework to deal with a complex scheduling problem that arises in a container terminal when multiple quay cranes must be assigned to holds of the same vessel to perform discharge (D) and loading (L) operations. In this problem (also referred to as the *quay crane scheduling problem* QCSP, (Legato, Mazza and Trunfio 2008)), the classical objective is to minimize the vessel's overall completion time (*makespan*), while taking into account precedence and non-simultaneity constraints between holds when they are operated by crane. According to our experience based on real data, D/L operation rates have been fixed to an average value of 26 containers/hour. For sake of sensitivity analysis, we have defined two D/L distributions for operation times: a realistic 8-order Erlang distribution and a basic exponential distribution.

Table 2: Comparison of Simulation Results Produced by Rinott's Procedure and Combined R&S Procedure

D/L Distribution	Solutions Evaluated	Average Makespan (hours)	
		RP	CP
Erlang	20	17.74	17.4
	135	14.05	14.45
	1379	12.93	12.67
Exponential	20	17.77	16.45
	135	14.15	14.24
	1379	13.04	13.21

Observe that the variance associated to the individual discharge/loading service times does not significantly affect the performance of the different procedures in terms of the average quality of solution returned by each of them (i.e. final, mean value of the makespan), as illustrated in Table 2. Rather, it has a clear impact on the average number of observations (simulation runs) required to achieve solutions, as one may recognize from results in Table 3 where the number of observations corresponding to the set of solutions in Table 2 are reported. In particular, the CP procedure seems to definitely outperform the classical, two-stage RP procedure by at least 20% whenever service times becomes less regular (exponential case).

Table 3: Comparison of Observations Required by Rinott's Procedure and Combined R&S Procedure

D/L Distribution	N° of observations		CP Performance (Δ%)
	RP	CP	
Erlang	212	209	+1.42%
	1421	1405	+1.13%
	14421	14283	+1%
Exponential	1100	791	+28.1%
	7035	5268	+25.12%
	71727	54029	+24.67%

5. CONCLUSIONS

We have discussed some recent issues on searching and ranking feasible solutions when using simulation-based approaches to optimization problems in logistics. This has been accomplished for the case in which candidate optimal solutions are all available at the initial step of the simulation-optimization procedure, as well as for the case in which they are revealed during the execution of the procedure. A subset-selection technique is shown to be effective for the problem of selecting the best yard crane assignment and transfer policy in a terminal container, due to the specific structure of the candidate policies. On the other hand, dealing with the quay crane scheduling problem, we obtained encouraging numerical evidence on the idea that changing the number of observations from a single one to multiple ones, according to the stability of the estimate of the process variance, may result as a practical key for speeding-up the simulation-based optimization procedure.

REFERENCES

Ahmed, M.A., Alkhamis, T.M., 2002. Simulation-based optimization using simulated annealing with ranking and selection. *Computers & Operations Research* 29:387-402.

Andradóttir, S., 1998. Simulation Optimization. In: Banks, J. ed., *Handbook of Simulation*. New York: John Wiley and Sons, Inc..

Andradóttir, S., Prudius, A.A., 2005. Two simulated annealing algorithms for noisy objective functions. *Proceedings of the 2005 Winter Simulation Conference*, 797-802. December 04-07, 2005, Orlando (Florida, USA).

Bartlett, M.S., 1937. Properties of Sufficiency and Statistical Tests. *Proceedings of the Royal Statistical Society Series* A 160:268-282.

Bechhofer, R.E., Santner, T.J., Goldsman, D.M., 1995. *Design and Analysis of Experiments for Statistical Selection, Screening, and Multiple Comparisons*. New York: John Wiley.

Canonaco, P., Legato, P., Mazza, R.M., 2007. An Integrated Simulation Model for Channel Contention and Berth Management at a Maritime

Container Terminal. *Proceedings of the 21st European Conference on Modelling and Simulation*, 353-362. June 04-06, 2007, Prague (Czech Republic).

Canonaco, P., Legato, P., Mazza, R.M., Musmanno, R., 2008. A queuing network model for the management of berth crane operations. *Computers and Operations Research* 35:2432–2446.

Canonaco, P., Legato, P., Mazza, R.M., 2009. Yard Crane Management by Simulation and Optimization. *Maritime Economics and Logistics* – Special Issue on OR Methods in Maritime Transport and Freight Logistics (in press).

Chen, E.J., Kelton W.D., 2005. Sequential selection procedures: using sample means to improve efficiency. *European Journal of Operational Research* 166:133-153.

Damerdji, H., 1994. Strong Consistency of the Variance Estimator in Steady-State Simulation Output Analysis. *Mathematics of Operations Research* 19:494-512.

Dudewicz, E.J., Dalal, S.R., 1975. Allocation of Observations in Ranking and Selection with Unequal Variances. *Sankhya* B7:28-78.

Gibbons, J.D., Olkin, I., Sobel, M., 1979. An Introduction to Ranking and Selection. *The American Statistician* 33(4):185–195.

Glynn, P.W., Whitt, W., 1991. Estimating the asymptotic variance with batch means. *Operations Research Letters* 10:431–435.

Goldsman, D., Meketon, M.S., Schruben L.W., 1990. Properties of standardized time series weighted area variance estimators. *Management Science* 36:602–612.

Goldsman, D, Kim, S.-H., Marshall, W.S., Nelson, B.L., 2002. Ranking and selection for steady-state simulation: procedures and perspectives. *INFORMS Journal on Computing* 14(1):2-19.

Gupta, S.S., 1965. On Some Multiple Decision (Ranking and Selection) Rules. *Technometrics* 7:225–245.

Hong, L.J., Nelson, B.L., 2007. Selecting the best system when systems are revealed sequentially. *IIE Transactions* 39.

Kim, S.-H., Nelson, B.L., 2003. Selecting the best system: theory and methods. *Proceedings of the 2003 Winter Simulation Conference*, 101-112. December 07-10, 2003, New Orleans (Louisiana, USA).

Laganà, D., Legato, P., Pisacane, O., Vocaturo, F., 2006. Solving simulation optimisation problems on grid computing systems. *Parallel Computing* 32(9):688-700.

Law, A.M., Kelton, W.D., 2000. *Simulation Modeling and Analysis*, Third Edition. New York: McGraw-Hill.

Legato, P., Mazza R.M., Trunfio, R., 2008. Simulation-based Optimization for the Quay Crane Scheduling Problem. *Proceedings of the 2008 Winter Simulation Conference*. December 07-10, Miami (Florida, USA).

Meketon, M.S., Schmeiser B., 1984. Overlapping Batch Means: Something for Nothing?. *Proceedings of the 1984 Winter Simulation Conference*, 227-230. November 28-30, 1984, Dallas (Texas, USA).

Milton, J.S., Arnold, J.C., 1986. *Probability and Statistics in the Engineering and Computing Sciences*. New York: McGraw-Hill.

Nemhauser, G.L., Wolsey, L.A., 1988. *Integer and Combinatorial Optimization*. New York: John Wiley.

Prudius, A.A., Andradóttir, S., 2004. Simulation optimization using balanced explorative and exploitative search. *Proceedings of the 2004 Winter Simulation Conference*, 545–549. December 05-08, 2004, Washington (DC, USA).

Prudius, A. A., 2007. *Adaptive random search methods for simulation optimization*. Ph.D. thesis. H. Milton Stewart School of Industrial and Systems Engineering, Georgia Institute of Technology, Atlanta, Georgia, USA. Available from <http://etd.gatech.edu/theses/available/etd-06252007-161154> [accessed April 10, 2008].

Rinott, Y., 1978. On Two-Stage Selection Procedures and Related Probability-Inequalities. *Communications in Statistics - Theory and Methods*. A8 799–811.

Song, W.T., Schmeiser, B.W., 1995. Optimal mean-squared-error batch sizes. *Management Science* 41:110–123.

Steiger, N.M., Wilson, J.R., 2002. An improved batch means procedure for simulation output analysis. *Management Science* 48(12):1569-1586.

Wilcox, R.R., 1984. A Table for Rinott's Selection Procedure. *Journal of Quality Technology* 16(2):97-100.

AUTHORS BIOGRAPHY

PASQUALE LEGATO is an Assistant Professor of Operations Research at the Faculty of Engineering - University of Calabria, where he teaches courses on simulation for system performance evaluation. He has published on queuing network models for job shop and logistic systems, as well as on integer programming models. He has been involved in several national and international applied research projects and is serving as reviewer for some international journals. His current research activities focus on the development and analysis of queuing network models for logistic systems, discrete-event simulation and the integration of simulation output analysis techniques with combinatorial optimization algorithms for real life applications in transportation and logistics. His home-page is <http://www.deis.unical.it/legato>.

RINA MARY MAZZA was born in Leyden Township, Illinois (USA) and went to the University of Calabria, Rende (Italy), where she received her Laurea degree in Management Engineering in 1997. She is currently pursuing a Ph.D. degree in Operations Research at the same university. She has a seven-year working experience on knowledge management and quality assurance in research centres. She has also been a consultant for operations modelling and simulation in terminal containers. Her current research interests include discrete-event simulation and optimum-seeking by simulation in complex logistic systems. Her e-mail address is: rmazza@deis.unical.it.

MODELLING, SIMULATION AND OPTIMIZATION OF LOGISTIC SYSTEMS

Pasquale Legato [(a)], Daniel Gullì [(b)], Roberto Trunfio [(c)]

[(a)] DEIS, Università della Calabria, Via P. Bucci 41C, 87036, Rende (CS), Italy
[(b)] CESIC – NEC Italia S.r.l., Via P.Bucci 22B, 87036, Rende (CS), Italy
[(c)] CESIC – NEC Italia S.r.l., Via P.Bucci 22B, 87036, Rende (CS), Italy

[(a)] legato@deis.unical.it, [(b)] daniel.gulli@eu.nec.com, [(c)] roberto.trunfio@eu.nec.com

ABSTRACT

In the last two decades, discrete simulation has become the most powerful tool in modelling logistic systems in a dynamic stochastic environment. An open challenge is trigged by the need to devise ways of developing easy-to-read and expressive visual modelling paradigms. Most modelling paradigms, as Event Graphs and Petri Nets, currently adopted mainly in an academic context, have been generally supplanted by simulation packages in real system applications. Nevertheless, our belief is that these are more expressive and powerful in modelling logistic systems. We rely on an innovative visual modelling paradigm based on the process interaction conceptual framework and on the holistic modelling approach, which we are presenting in this paper. The paper focuses on the way of representing processes, resources and entities that compose our simulation modelling paradigm. Modelling capabilities of our modelling paradigm are compared to those of Event Graphs and Petri Nets within a real case study.

Keywords: modelling, simulation, optimization, logistics, container terminal

1. INTRODUCTION

In a stochastic dynamic environment, discrete event simulation (DES) models are well capable of representing the behaviour of large and complex logistic systems. Thus DES models are widely adopted as planning and control tools to estimate system performances under uncertainty and conduct scenario analysis.

Modern, commercial DES simulation packages based on a "point & click" logic – that is what Pidd (2004) calls VIMS – are devoted to minimize the system modelling effort by hiding the behavior of the components adopted to construct the model, but at a loss of model readability, understanding and customization. In opposition, a *modelling paradigm* (MP) is a visual modelling approach based on a formalism designed on a worldview which requires a lot of modelling ability, and which provides superior representational capabilities.

In the past, a lot of interesting and powerful DES modelling paradigms (MPs) have been developed based on the three classical worldviews (or *conceptual frameworks*), i.e. *Event Scheduling* (ES), *Activity Scanning* (AS) and *Process Interaction* (PI) (Derrick et al. 1989). The most notable MPs developed using these conceptual frameworks are respectively: *Event Graphs* (Schruben 1983), *Petri Nets* (Petri 1962) and *Hierarchical Control Flow Graphs* (Fritz and Sargent 1995). Derrick et al. stated that a worldview is an underlying structure and organization of ideas which constitute the outline and basic frame that guide a modeller in representing a system in the form of a model.

The PI conceptual framework better suits the needs for model readability and understanding. In fact, while ES and AS modelling approaches are based respectively on events and activities, the PI is based on the concept of *process*. A process is a complex concept which represents a flow of events and activities through which a particular model object moves; therefore, in a model specification, it describes the life cycle of a model object. Whilst a model object moves through its process, it may experience certain delays and be hold in its movement. Thus, the first thing to do in the PI worldview is to identify all the model objects which are involved in the model specification. Subsequently, the modeller must specify the sequence of events and activities for each model object. For this reason, a process is often represented with a schematic representation known as *flow-chart*, where the events and activities that compose the process are the nodes of the chart. A flow-chart is usually represented as a directed graph. For each activity included in a process routine there is a stretch of simulated time (even null). The PI qualities are a moderate burden for the modeller, a high maintainability, an excellent natural representation capability. The most notable drawback is the high burden on execution, even if modern object-oriented approaches better fit the implementation needs of this conceptual framework, and the effort required for the developmental time, which is quite high.

Modelling a stochastic system using a MP or, in alternative, a commercial VIMS (e.g., GPSS and Arena) implies the use of a specific modelling approach. The two classical dichotomous modelling approaches are *reductionism* and *holism*. Reductionism relies on the belief that a complex system may be decomposed into its constituent parts without any loss of information, predictive power or meaning (Pidd and Castro 1998). The totality of MPs and VIMS are developed using this modelling approach. Unfortunately reductionism does not consider that large and complex logistic systems cannot be decomposed into their constituent parts with the guarantee of all the above conditions. This statement can be practically proved by trying to analyse real systems, e.g. *supply chains* and *container terminals*: in the latter case, it is easy to verify that it is not possible to decompose the system into its constituent parts (logistic processes) and analyze them separately, because they are partially or entirely related: some significant loss of information, predictive power and meaning will occur. Pidd and Castro (op cit) have shown that the best approach for the management of a complex system is based on holism. Holism assumes that systems possess some properties that are meaningful only in the context of the whole and not in the parts (e.g., in a maritime container terminal, shuttle vehicles must be assigned to the quay cranes considering the whole work-load). Therefore, using a holistic approach to model a whole system achieves better results in terms of the replication of the real system behaviour. Thus, developing an MP or VIMS by using the holistic approach should be appropriate. However, considering that both alternatives are affected by an intrinsic difficulty in model understanding, which makes simulation models unpopular at the model end-users, here we propose an MP which provides: *i)* a high representational capability from the conceptual point of view, as well as *ii)* a common language for both modellers and end users from the model understanding sight.

We believe that a stimulating possibility is to define an innovative MP for modelling logistic systems under uncertainty using a holistic approach, having in mind the goal of developing a simulation based optimisation platform. Besides, the considerations about strong and weak points of the ES, AS and PI worldviews provided by Derrick et al. (1989) convince us that we must define our MP using the PI conceptual framework. Thus, we dedicate the next section describing our MP, based on a previous work (Legato and Trunfio 2007). Successively, to compare our MP with two different, successful MPs, EG and PN, we model the same real logistic process in a container terminal, showing the different way of modelling this reality by using these MPs. Finally, we discuss some specific issues when developing a friendly tool for the integration of optimisation techniques within simulation models.

2. WHOLISTIC DISCRETE EVENT SIMULATION WITH PROCESS INTERACTION

In this paper we propose a holistic MP for DES modelling based upon the PI, that we call *Holistic Modular Process* (HMP) simulation models. In the following, we define the main concepts of our simulation MP. In the subsequent section we illustrate its potentiality by modelling a typical logistic process that arises in a maritime container terminal and comparing it with the EG and PN models.

2.1. Holistic Modular Process Simulation Models

The MP proposed in this paper is aimed to be flexible and expressive in the modelling of complex systems. It tries to achieve three primary objectives: model readability, reusability and customizability.

Model readability is a property which allows a model to be simple-to-read for a non-modeller. This property has a special importance when top managers are directly involved in scenario analysis: in our MP readability is achieved by describing the components' behaviour within a simulation model by a sort of flow-chart. As for reusage property, our experience at the Gioia Tauro Container Terminal confirms the requirement that a specialised simulation tool has to be reused in some of its forms (model reuse, component reuse, function reuse and code scavenging). Model reuse, under calibration and repeated tuning, occurs as soon as traffic conditions change over time. Furthermore, component and function reuse are both required to give the operational manager the possibility of quickly implementing a first order model of some emerging situations, before the structured intervention of external expertise. According to our concept of holistic MP, we provide model reusability by means of hierarchical, modular model definition and redefinition of simulation parameters.

Model customizability is the base of an MP for the effective modelling of complex systems. It relies on the user-definition of process properties that allow describing uncommon situations, as it is the case when the modeller is asked to represent local, best practices in logistics organisation and management.

An outline of the HMP simulation models follows now. An HMP model includes a set of *model objects*, or *objects* for short. For each model object an inner and outer view is defined. The outer view is depicted as a box equipped of input and output ports (see Figure 1). The inner view depicts a sequence of activities and events. The sequence is also called the *model object process routine*, or simply *process*.

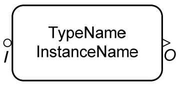


Figure 1: The Outer View of a Model Object in a HMP Simulation Model.

In a similar way as described for HCFG models, relationships between model objects are depicted using directed edges (*channels*). Model objects are equipped with *input* and *output ports* (depicted as shown in Figure 1) that can, eventually, be renamed to improve model readability. Two model objects are linked by means of only one channel from an output port of the first object to an input port of the second object and vice versa, i.e. no more than two channels connect directly two model objects. Two model objects interact by message passing via channel, despite a channel has its own direction (the head of the edge is connected to an output port, while the tail to an input port). Multiple types of messages flow forward and backward along a channel, whilst *entities* (or *jobs*) of the simulation model can flow only through the proper channel direction.

In this way, by adopting a modular approach, an HMP simulation model is a net of model objects. Therefore, hierarchical modelling is pursued by coupling different processes and grouping the resulting net of model objects into a *sub-model* (which is depicted in a similar to a model object, as proposed in Figure 2). In this case, the sub-model can be used to be coupled together with other model objects or sub-models. A requirement when constructing sub-models is that at least one input or output port must be defined and linked to the inner model-objects, otherwise, the depicted model is a *super-model* or *final model*. The inner view of a sub-model is just another HMP model where some channels link the inner part to the outer part of the model through its boundaries.

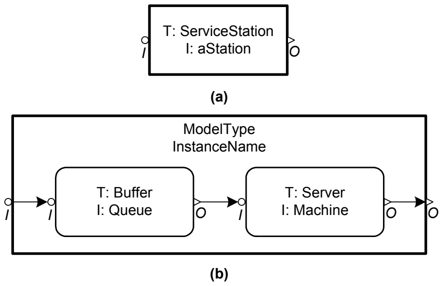


Figure 2: (a) The Outer View of a Sub-Model depicting a Service Station. (b) The Inner View of a Service Station, Composed by a Buffer and a Server.

As stated above, the inner view of a model object is a process, or rather a sequence of activities and events that define the model object behavior. The process is represented as a particular hierarchical flow-chart, called *Event-Activity Diagram* (EAD), where activities and events are nodes and edges fix the logical and temporal sequence between nodes (other node types will be introduced in the following). A process can be constructed hierarchically by grouping events and activities to compose a sub-process. Thus, a sub-process is an EAD itself. A sub-process we provide component reuse and model readability.

A detailed model definition is obtained by the EAD. An EAD provides the process definition. In fact, an EAD defines the structure of a process by means of a sort of flow-chart. Because of the possible use of sub-processes for the process definition, an EAD could be a hierarchical structure (composite nodes are expanded revealing a new EAD) and a hierarchical tree may be used to explore the EAD structure.

2.2. Model Objects

Following the PI worldview, the first step consists in identifying all the model objects which are involved in system modelling and detect all common features. Then, class-dependent features are described for the few classes of model objects introduced in our MP. The objects are contextual, so it is necessary to specify the model in which they are defined. The *model name* parameter is used to declare which model an object belongs to. This parameter serves two reasons: first of all, the holistic approach says that such objects can only be used in some contexts; furthermore, the use of sub-models could cause a lot of confusion, especially if a sub-model is exploded (i.e. deleting model boundaries) and the same object is used in a model and in its sub-models. Objects of the same type are identified by the *type name* parameter. If the model name is a parameter that depends on the model, the type name is a non-changeable parameter. Each instance of a certain type of object is also identified by the *instance name* parameter. The set composed by these three parameters univocally identifies a model object within an HMP simulation model.

There is at least another important parameter that allows managing heterogeneous objects, known as *category name*. The use of the category name allows us to make associations between objects that are apparently disjoined. A possible use can be seen in the development of a simulation package, in the packages for statistical analysis and optimization of system performance measures. As a matter of fact, in this context generic rules and algorithms can be defined over a class of mixed objects.

Model objects have a set of variables and data structures used to support the logical representation of the process behavior. These properties are not explicitly depicted and refer to the code implementation of each model object type. Model objects variables and data structures are accessed by a set of public functions which allow their manipulation. A function is called by message passing, or rather sending an explicit message to call a specific model object function.

Model objects are illustrated in detail in the following.

There are two basic classes of objects: *resources* and *resource managers*. As stated above, entities of the simulation model are depicted as messages that are able to envelop properties, data structures and other entities (e.g., a ship that carries thousands of containers loaded in different holds).

Resources are *active* or *passive* depending on their role in the simulation model. Passive resources are not depicted explicitly and are not able to execute action/events or process entities. Nevertheless, a passive resource is able to execute incoming requests and actions of other model objects (as declared above, a passive resource shows a set of public functions that can be used to manipulate the resource). Passive resources are generally managed by an active resource or a resource manager. Whenever a passive resource is managed by a resource manager, active resources linked to the resource manager are able to overwork it only under the conditions specified by the resource manager. A passive resource must be managed by a resource manager if more than one resource may request it for use during a simulation (e.g., items storage into a shelf using forklifts). An active resource can possess passive resources and it can offer a service to one or more entities per time. It can also make queries to other objects to which it is linked by message passing via input/output ports. The behavior of an active resource is described using an EAD.

By means of resources one can only represent just a system governed by a few simple rules. As matter of fact, the need of modelling complex systems in a holistic approach leads us to introduce the resource managers. A resource manager is a high-level object, which is able to interact with a set of model objects (also heterogeneous). Resource managers can take decisions (e.g., solve a scheduling or an assignment problem, negotiating the use of a sub-system, etc.) by applying rules and policies and making queries to other model objects. They have free access to modify the behaviour of all the resources in their own model to which they are linked. In a holistic approach, as demonstrated by Pidd and Castro (op cit), the use of that kind of model objects avoids the explosion of object links and therefore it represents a more easy-to-use modelling tool.

2.3. Processes and Event-Activity Diagrams

The role of a process in model object specification is analysed here and process representation is shown. The interaction of processes during the simulation is briefly described.

According to the definition of process given in the PI worldview overview, a process is a series of temporally related events and activities. Usually flow-charts may be used to represent processes. In our flow-charting methodology, called *Event-Activity Diagram* (EAD), events and activities are nodes, while directed edges define one or more paths that can be covered by a process. Other useful elements compose a process, namely the *logical nodes*.

Activities and events are not intended to perform actions, but to show to non-modellers, in a friendly-way, how a process can work. The role of executing requests, performing actions and introducing time delays between activities and/or events is assigned to directed edges. Therefore, the use of a flow-charting graphical methodology to depict a process allows us to achieve at a good extent the readability objective. Bearing in mind the list of the process components, let us start an in-depth discussion about these components.

In our MP, activities are classified focusing on the simulation duration of the activity; hence activities are partitioned in *timed* and *instantaneous*. The first type of activities are those able to start operations at a simulated time instant and finish operations in a future simulated time instant. For instance, timed activities are those that perform operations characterized by a variable simulated time length, e.g. waiting or servicing activities. The second type refers to activities that start and end operations at the same simulated time instant, e.g. activities representing a choice or check by a resource or resource manager. Nevertheless, also timed activities can starts and ends operations at the same simulated time instant.

A process needs the specification of an initial activity that is enabled, or rather the activity that possesses the process checkpoint (more details about checkpoints are provided in the following). The initial activity is the activity from which the process starts when it becomes active (initial process state). In an EAD one or more activities per time can be enabled, i.e. when the process flow has been forked in different logical paths. The set of currently enabled activities is called the *process state*.

An event is a fact that forces one out of a set of possible changes of the current state. It may precede or follow an activity, thus representing something that is just happened or that is going to happen. If an event precedes an activity, then it is processed at the same simulated time of the activity start; if an event follows an activity, then it is processed at the same simulated time of the activity end.

As stated before, EADs have a hierarchical process structure. In fact, nodes are activities and events and even sub-processes. A sub-process is itself an EAD, therefore it can be zoomed revealing the included flow-chart. The EAD of a sub-process can refer to input and output ports of the including process, i.e. a sub-process is only a convenient arrangement for grouping an EAD sub-net and depicting it as a single node (this solution aims to improve model readability).

The different shapes for the main nodes of an EAD are depicted in Figure 3.

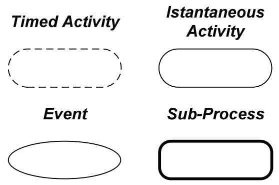


Figure 3: EAD Activities, Events and Sub-Processes Nodes.

The core of the process behavior is designed adding directed edges. Edges are used *i)* for message passing and the evaluation of conditions through *functions* call, and *ii)* for the introduction of timing into process activity flows. We classified edges in four types: *null edges* (or *true edges*), *inner edges*, *incoming edges*, *outgoing edges*. A null edge can be used only to connect events to activities or activities to activities. An inner edge connects an activity to another activity or to an event. These edges are depicted as arcs (see Figure 4). Incoming edges are used to depict incoming messages from an input/output port, while outgoing edges depict outgoing messages to an input/output port. Incoming edges connect only activities to activities or activities to events, while outgoing edges also link events to activities. Edges may not have any node linked to the head, thus they work as a termination arc (i.e, the process becomes definitely terminated when one of these nodes is traversed).

Edges produce a transition from the current enabled node (connected to the edge tail) to the future enabled node (that is linked to the edge head). To avoid deadlock, edge tail and head must be different nodes. While the transition rule for a null edge is always true, for the other edge types the edge traversal is allowed only if certain conditions are met. To understand the nature of these conditions, we introduce the following edge functions: a *Timer* and a *Request*.

A Timer is a time function that generates a delay time using a distribution (e.g., exponential) and the appropriate set of parameters. Only inner and outgoing edges may have a Timer. Whenever an inner/outgoing edge is selected by an activity to be eventually traversed, the Timer generates the simulated delay time and an "alarm" is scheduled to warn the edge after the delay time. When an edge has been warned, it is allowed to enable its head node (i.e., the process state has been changed), or rather it is traversed.

The transition rule of an edge is composed by a set of Requests. A Request is a (public or private) function of a process that can be called to check conditions, to set up model objects and entities parameters and (in case) exchange entities among model objects. If the conditions are met, the function returns true; false, otherwise. A set of primary Requests can be used on the same edge by the formulation of a logical expression using boolean operators (*and*, *or*, *xor*) and negation (*not*), that we call the *primary rule*; moreover, the verification of the primary rule may depend on a set of conditioning Requests, or what we call the *conditioning rule*. Both conditioning and primary rules compose the transition rule in the following way: "primary rule | conditioning rule". Only if the conditioning rule is evaluated as true, then the primary rule is evaluated.

All the edge types (excepted null edges) may have a transition rule. A considerable difference exists between inner and incoming/outgoing edges. In fact, by using incoming/outgoing edges, a process can: *i)* make a call to a public function of an external process (*receiver process*) linked to an input or output port; *ii)* receive a call to its own public functions by an outer process (*sender process*) linked to an input or output port. In this way we enable communication among processes (e.g., exchanging entities and assigning work). In particular, the primary rule of an incoming (outgoing) edge must only include the call of a Request by (of) another process linked to an input/output port. The symbolism used to call a Request of an outer object is "PortName<Request", while to depict the call of a Request by an outer object "PortName>Request" is used. For incoming/outgoing edges, the conditioning rule may also include calls to outer functions.

If a sender process calls a function of a receiver process, and this occurs only if at least an incoming edge starting from the current node of the receiver has the called Request as primary rule, then the call to the Request function may return true; otherwise, false is automatically returned (i.e. the call is not accepted). Indeed, incoming edges work as triggers for processes that are waiting for an external input.

With the exception of null edges, which are always traversed, if an inner edge (or an outgoing edge) is selected to be traversed by the current node, then it is traversed once the traversal rule is true.

Timers and Requests are the condition functions that may be involved to cause an edge traversal. Nonetheless, inner and outgoing edges may have both a Timer and Request function. In this case, the execution priority states that the Timer must be executed before a Request, i.e. the transition rule is verified when the edge has been warned.

All the edges may also have a list of *Action* functions. An Action is a function that performs such operation, e.g. changing the parameters of the process or of an owned entity. If a Timer and/or a transition rule have been defined for the same edge, the Action must always be executed at last, i.e. after the edge has been warned and if true has been returned by the transition rule.

Each Request and Action can explicitly receive a set of parameters; therefore, the behavior of a Request/Action may change in function of the received parameters.

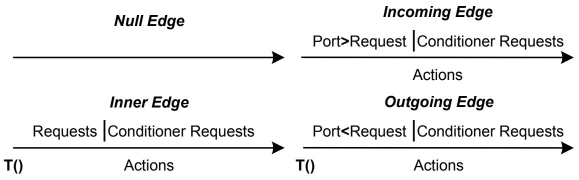

Null Edge
Incoming Edge
Port>Request |Conditioner Requests
Actions
Inner Edge
Outgoing Edge
Requests |Conditioner Requests
Port<Request |Conditioner Requests
T()
Actions
T()
Actions

Figure 4: Edges of an EAD.

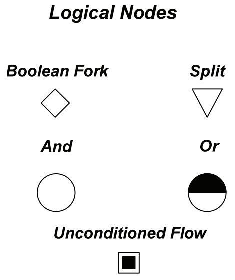


Figure 5: Logical Nodes of an EAD.

To improve MP scaling and model object reusability, as proposed in Sargent (1997), each process may use *multiports*. A multiport is an indexed set of *k* ports named *portname[1],..., portname[k]*. A multiport is explicitly depicted as different ports in the external view of a model object; in the inner view, multiport may be used either in an explicit or implicit way by incoming/outgoing edges. For instance, an explicit use of a multiport may occur when the process needs to call a Request function of an external process linked to the port "*portname[i]*" (where the suffix *[i]* stands for the index of a port that is included in the multiport *portname[1,…,k]*), then the name of the port is explicitly depicted in the transition rule of the function as follows: "*portname[i]<Request*". Another example is introduced to show the implicit use of a multiport. If the process needs to call the same Request function of the external processes linked to the multiport, the following notation must be used in the transition rule of the outgoing edge: "*portname[i:1,…,k]<Request*". In this way, for each *portname[i]*, where index *i* varies between 1 and k, the edge must check the associated transition rule and the edge is traversed whenever at least one transition rule is true. The implicit use of a multiport allows our MP to be compact and to easily implement the behavior of dynamic processes. Moreover, by using implicitly a multiport, it is possible to specify a sub-set of port indexes that *i)* must be called through an outgoing edge or *ii)* allow an external process to call an inner Request through an incoming edge. The sub-set of port indexes can also be a list name generated at runtime using an Action function.

Our MP allows defining a process behavior via EAD using activities, events, sub-processes and directed edges. Another type of nodes, called *logical nodes*, has been defined in order to support the definition of process paths. These nodes are: *i) boolean fork*; *ii) split*; *iii) and; iv) or*; *v) unconditional flow* (Figure 5). A logical node may be used to represent alternative paths to be chosen under specified conditions.

A boolean fork can be connected to the head of a conditional edge, or rather those edges that have a transition rule. Using this node, if the transition rule is true, than the edge is traversed and the process control flows through an edge starting by the boolean fork node; otherwise, the edge is also traversed, but the process control flows through a special edge, depicted with a dashed line, which starts by the boolean fork node. Therefore, using the boolean fork node after a conditional edge, the edge is always traversed. The outgoing edges from a boolean fork node are of the null or inner type. In case inner edges are used, another boolean fork must be used for each inner edge to catch the result of the transition rule.

The unconditional flow node acts in a similar way of a boolean fork, but whatever be (true or false) the transition rule of the incoming edge, only one edge must leave this node.

The remaining logical nodes may be connected to the head of any edge type. Once the edge is traversed, the process control passes at the logical node. A split node is used to separate the process path in two or more alternative paths. When the process path is separated in more paths, to recombine two or more paths, an and/or node is required. The and node, becomes enabled at the time all the edges incoming to the node have been traversed. The or node become enabled at the time at least one of the edges incoming to the node has been traversed.

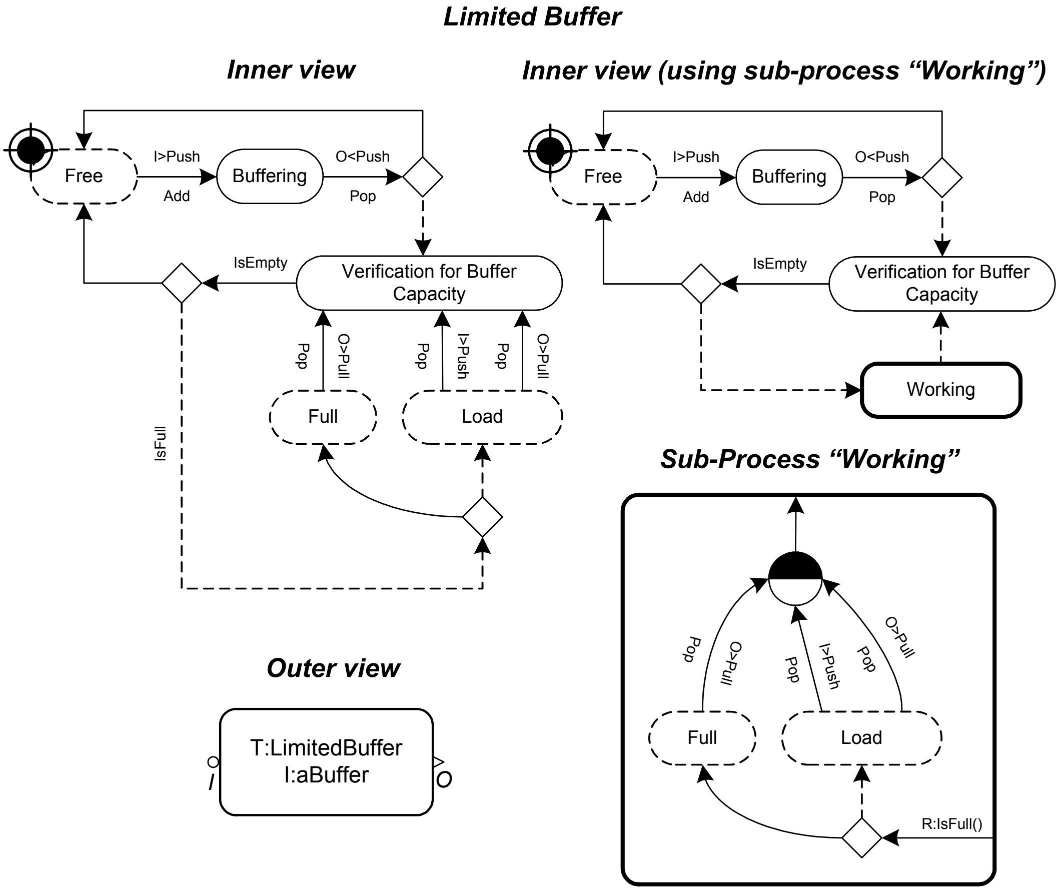


Figure 6: An Example of Use of a Sub-Process for a Model Object that is Designed as a Buffer with a Limited Capacity.

The and, or and split nodes can be part of the process state. Typically the process state has only one active activity per process state. However, anytime the process flow is separated by using a split, many activities and the split/and/or nodes may be enabled, thus they may be part of the process state.

Some explanations are required about the use of sub-processes as nodes of an EAD. As they are EAD nodes, edges that start and arrive to these nodes are of the types defined in the previous. The only restriction is for edges that start by a sub-process node. If an edge that starts from a sub-process node is not a null edge type, a boolean fork must catch the edge transition results (otherwise the process checkpoint may stay on an edge included in the sub-process). The inner view of a sub-process is a particular EAD which consists of at the most one edge that links the outer view to an inner node (or rather from the inner boundary to a node) and/or at the most one edge that links an inner node to the outer view (or rather from a node to the inner boundary). Also if the edge that starts from the sub-process inner boundary to a sub-process node is not of null edge type, a boolean fork must catch the edge transition results (otherwise the process checkpoint may stay on an edge included in the sub-process). An example about a limited buffer is shown in Figure 6 (details of the Request and Action functions used in this model object are provided in a PhD thesis).

Processes are usually executed during more simulation time periods. For this reason the nodes of a process can be visited during different time periods. To track this possibility, we adopt the *process checkpoint*. A process checkpoint is a property of the processes that shows the set of nodes from which a process starts or is reactivated, i.e. the checkpoint locates the enabled nodes (the process state). A process checkpoint is also called *reactivation point*. For each process, the reactivation point must be explicitly shown to depict the initial active states as done in Figure 6 for the "Free" node (that is the initial state of the buffer model object): the reactivation point is depicted as a "target" and is usually placed on the upper-left corner of a node.

In HMP simulation models, a process can have the following status in the sense of simulation execution: *active*, *passive* or *terminated*. Focusing on a non-concurrent simulation, only one process can be active at a time. A process is active in the sense that it is moving through its paths, until it enters the passive state or is terminated. A process enters the passive state when a Timer schedules a delay time or a message is sent to an input/output port. In a similar way, a process becomes active by means of an external trigger (a message from an input/output port) or a previously scheduled warning time. To start model simulation, at least one process must receive a message to be activated for the first time. An active process may become terminated whenever an outgoing or transfer edge is traversed and the edge head is not linked to any node. A terminated process returns false to each incoming message and will never be active during the simulation.

3. A MODELLING CASE STUDY

In this section, a real case study of modelling and optimization of a complex logistic process is presented, with the aim of comparing the expressiveness of our MP with other well-known approaches.

The system that we are intended to model concerns the management of quay crane operations and it includes some key logistic processes, such as the "vessel discharge/loading" and the "container transfer from ship to yard (and vice versa)". This is a typical operative management problem that requires both simulation and optimization to be successfully approached. We refer to the Gioia Tauro terminal, one of the major container terminals in the Mediterranean Sea, located in Southern Italy.

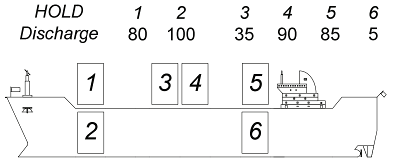


Figure 7: Map with Discharge/Loading Info per Vessel Hold

To fix ideas, we give a brief description of the system which includes the quay crane operations problem. Once a vessel is moored in a maritime container terminal, a certain number of containers must be discharged/loaded from/into the vessel's holds according to a pre-established operation plan (an example is shown in Figure 7).

Container discharge/loading can be initiated only if mechanical (and human) resources are allocated; if not, the ship waits in its berth position until resource assignment. Discharge/loading operations are performed by Rail Mounted Gantry Cranes (RMGCs) placed along the berth: one or multiple cranes move containers between the ship and the quay area. When multiple cranes are assigned to the same ship, crane interference has to be avoided and a complex scheduling problem arises to manage the relationships (precedence and non-simultaneity) existing among the holds of the same ship (this is the "Quay Crane Scheduling Problem" - Legato et al. 2008).

Table 1: Events.

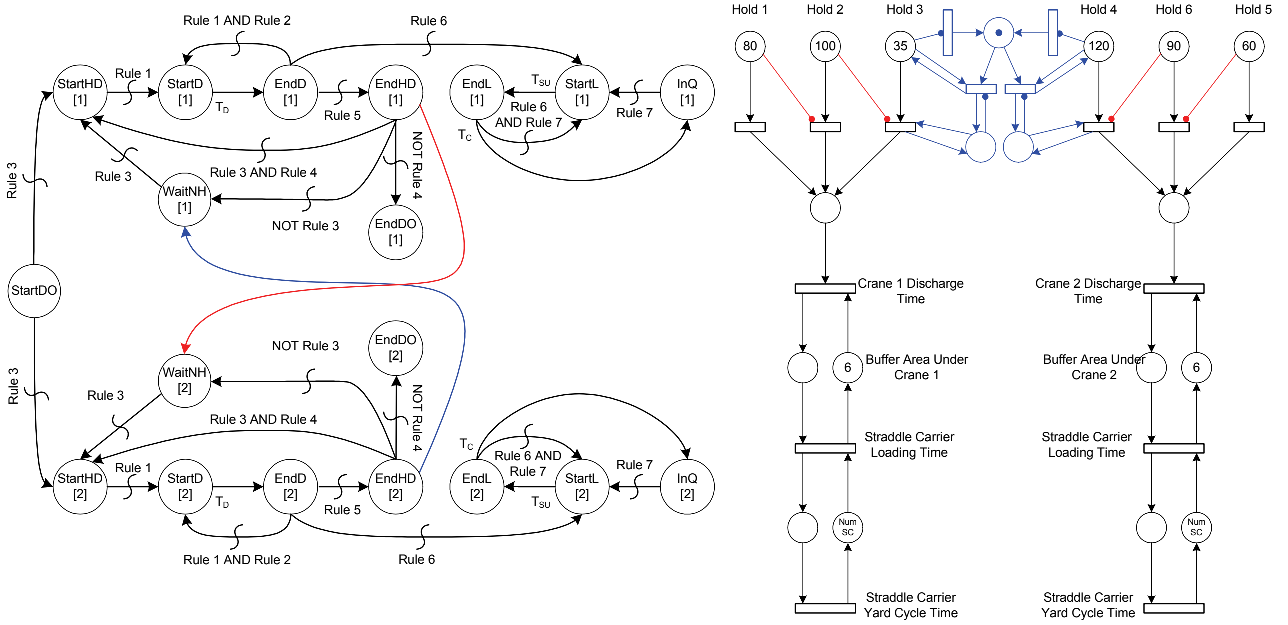
Event	*Description*
StartDO	Cranes start discharging operations
EndDO[i]	Crane i-*th* ends assigned discharging operations
StartHD[i]	Crane i-*th* starts discharging of *current hold*
EndHD[i]	Crane i-*th* ends discharging of *current hold*
StartD[i]	Crane i-*th* starts discharge of a container from the current hold
EndD[i]	Crane i-*th* ends discharge of a container from the current hold
WaitNH[i]	Crane i-*th* waiting to acquire the next hold

InQ[i]	Straddle carrier arrival in quay
StartL[i]	SC starts container loading
EndL[i]	SC completes container loading

Table 2: Enabling conditions.

Edge Condition	*Description*
Rule 1	Buffer area under crane capacity is less than 6
Rule 2	Hold has not been completely discharged
Rule 3	No violation of precedence and non-simultaneity constraints
Rule 4	Crane has more holds to be discharged
Rule 5	Current hold is completely discharged
Rule 6	At least one SC waiting in quay
Rule 7	At least one container under the crane

The transport of containers from the quay side to the yard side and the reverse movement depends on the transfer mode of the container terminal. At Gioia Tauro, containers are transferred according to the direct transfer mode, i.e. the transport of containers and the container stacking processes are performed by a fleet of straddle carriers (SCs). SCs take in charge containers and cycle between the berth area and the assigned storage positions within the yard. SCs are guided to a slot within the yard structure to stack/retrieve containers. They are also able to cover the distance from the assigned RMGC to the yard slot and back.



(a) **(b)**

Figure 8: (a) The Event Graph of the Quay Crane Operations Model. (b) The Petri Net for the Quay Crane Operations Model.

Here, we are interested in depicting the container discharge process of a certain vessel and depicting the container transfer from the limited buffer area (max 6 containers) under a RMG quay crane to the yard side by means of SCs.

Since the process of transferring containers from the quayside to the yardside is performed by SCs that are used to cycle among these areas with (outward path) and without containers (backward path), this process may also be depicted as a unique service time, or rather “the yard cycle time”, which includes the outward time, the container unloading time and the backward time.

In the following, we suppose for simplicity that two RMGCs must simultaneously perform the discharging operations of the vessel depicted in Figure 7. The sequence of holds that must be respectively discharged by the two cranes is the following: {*Hold 1, Hold 2, Hold 3*}, and {*Hold 6, Hold 5, Hold 4*}.

Figure 8(a) shows the Event Graph model of the logistic processes described above; events are labeled with numbers in squared brackets that refer to a specific crane.

Tables 1 and 2 report the descriptions of the events (the vertices of the graph) and the edge conditions. The edge delay time functions T_D, T_{SU}, T_C, are respectively: *i)* container discharging time for each RMGC, *ii)* container set-up time for a generic SC, and *iii)* cycle-time for an SC (i.e. the time to transfer a container from the quay to the assigned yard-slot, to set-down the container and, finally, to move back to the assigned RMG quay crane). As a matter of fact, the possibility to discharge a hold by a crane is due to the respect of the precedence and non-simultaneity constraints. In this way, the availability of a hold depends on the current task that is assigned to the next crane. In our example, if crane 1 is discharging hold 3, then crane 2 to discharge hold 3 must wait the completion of hold 1 (and vice versa). To depict this process, every time a crane completes its discharging operation on a hold, then it warns the next crane (edges in blue and red).

Using Petri Nets, we can model the system as depicted in Figure 8(b). In this model, seeing that the number of tokens in some places were too large to be explicitly depicted (e.g. the tokens which stand for the containers to be discharged) we have directly put in each place the number of tokens. Using Petri Nets, we are able to design the non-simultaneity constraint between Hold 3 and Hold 6 (edges and nodes in blue), as well as precedence between holds to be discharged (inhibitor arcs in red). Nevertheless, we do not provide reasonable model readability, and the net explosion is inevitable every time the number of holds and constraints rise up, as in real cases.

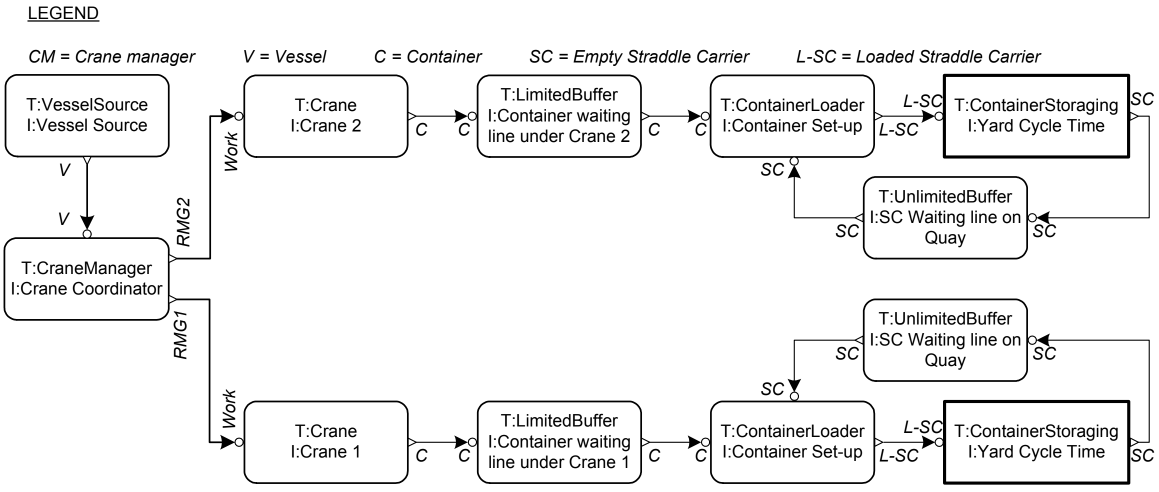

LEGEND
CM = Crane manager
V = Vessel
C = Container
SC = Empty Straddle Carrier
L-SC = Loaded Straddle Carrier
T:VesselSource I:Vessel Source
V
V
T:CraneManager I:Crane Coordinator
RMG2
RMG1
Work
Work
T:Crane I:Crane 2
T:Crane I:Crane 1
C
C
T:LimitedBuffer I:Container waiting line under Crane 2
T:LimitedBuffer I:Container waiting line under Crane 1
T:ContainerLoader I:Container Set-up
L-SC
T:ContainerStoraging I:Yard Cycle Time
SC
T:UnlimitedBuffer I:SC Waiting line on Quay

Figure 9: The HMP for the Quay Crane Operations Model.

Finally, we provide a representation of the quay crane operations model using an HMP model, as shown in Figure 9, based on the coupling of six types of model objects and a sub-system. The model objects are: *i)* a *VesselSource* for the generation of vessels, *ii)* a *Crane* for depicting the quay RMGC resource, *iii)* a *CraneManager* to manage the interactions among RMGCs working on the same vessel and assign the sequence of holds of the berthed vessels, *iv)* a *ContainerLoader* (Figure 10) to perform container set-up operations for empty SCs, *v)* an *UnlimitedBuffer* (Figure 6) to depict the waiting line of empty SCs that wait for load discharged containers and *vi)* a *LimitedBuffer* to depict the buffer area under a crane (for no more than 6 TEUs). The EADs for the *UnlimitedBuffer*, *VesselSource*, *Crane* and *CraneManager* model objects will be provided in a companion paper.

With this approach, a modeller can design the behavior of a set of context specific model objects (e.g., cranes and queues), even unrelated, and then he/she can reproduce a certain system by linking these model objects and simply specifying suitable parameters (e.g., the distribution function for the crane discharge time). If the designed model objects are collected in a library in a "point & click" simulation environment, they are close at hand to be reused to design a new system.

It is clear that by using this approach we provide a compact way of representing the system logic: in fact, entities flow through the model objects and the relationships between the model components are very clear. This is true especially if the model end-user looks at name of the ports, which explicitly indentifies which are the input(s) and the output(s) of each model object.

Moreover, if one wishes to further simplify the readability of the model by hiding a given part of the system, this can be performed in a simple way by grouping and depicting the selected system part as a sub-model. For instance, in Figure 9 the *ContainerStoraging* is a sub-model that receives in input a loaded SC and returns an empty SC after a while (the yard cycle time); focusing our attention on the quayside operations, we are not interested in understanding how an SC has been unloaded and why it has returned after a certain time period, therefore the use of a sub-model in this context appears convenient.

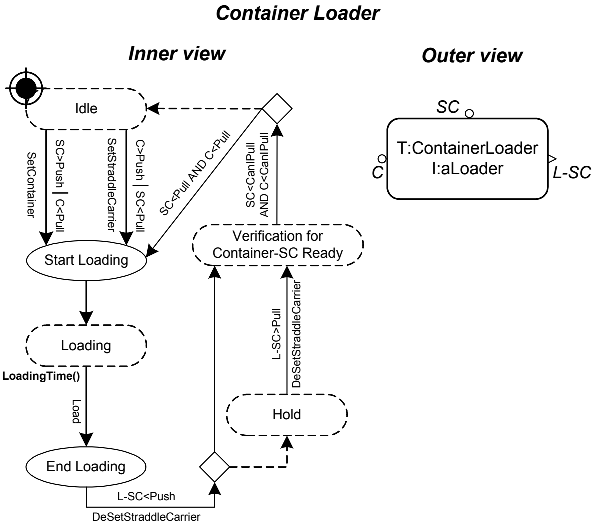


Figure 10: EAD for the ContainerLoader model object.

The behavior of each model object is designed by using an EAD. Each EAD is composed by *i)* default components (events, activities, logical nodes, edges) and *ii)* context specific timers and functions. Both timers and functions are developed around a set of global variables of each process (e.g., a list of entities for the infinite queue depicted by the model object

UnlimitedBuffer) and using a scripting language like TCL (www.tcl.tk).

This approach has its strong point on the use of functions to check conditions and execute algorithms devoted to take decisions. For instance, once a vessel arrives at berth, the CraneManager uses a function that solves a mathematical model (the Quay Crane Scheduling model proposed by Legato, Mazza and Trunfio – 2008) to dynamically assign the holds of the vessel to a specific set of cranes. Obviously, this approach is possible using our MP or EGs and HCFGs. Petri Nets are a powerful tool for modelling a system at a very low level, describing also conditions and complex constraints by means of its intuitive formalism, but a PN cannot be re-designed at runtime in case of need (e.g., is not possible to use the PN in Figure 8(b) to simulate the discharge process of a vessel with a different structure).

4. SIMULATION-BASED OPTIMIZATION

A modern simulation platform for logistic systems must provide an easy-to-understand language for system modelling, but also a tool for designing of optimization problems. We are developing a platform for the optimal management of logistic systems management based on the following three phases: *i) system modelling*, *ii) model analysis* and *iii) system optimization*. In the first phase, the real system is modelled by using our MP, as discussed in-depth in the second and third section.

Numerical results from the simulation model obtained in the system modelling phase are analyzed in the second phase by means of a statistical analysis tool. Performance measures are evaluated on the outputs from the simulation runs of the simulation model. In the model analysis phase, indices and parameters are defined on the simulation model by means of some tool or language. The evaluation of these indices and parameters provides the set of performance measures necessary for the simulation-optimization platform.

After that, in the third phase an optimization problem is defined. General problem setting is made of input and output variables, objective function and constraints. The nature of the optimization problem is intrinsically stochastic, due to the nature of the simulation output variable). Output variables are simulation model performance measures; input "variables" are quantitative or qualitative in nature. In this context, a language to easily develop simulation-based optimization algorithms must be planned, e.g. by using the TCL language.

5. CONCLUSIONS

We have proposed a Modelling Paradigm to support the development of simulation models oriented to the optimal management of logistic activities. The MP uses a holistic approach to capture the complex relationships among sub-systems and allows for a direct representation of the hierarchical structure of the decision making process for system management. System modelling is completed by a flow-chart based definition of processes involved in the model at hand. Real case examples of modelling have been presented, comparing our approach to those provided by Event Graph models and Petri Nets, with the aim of supporting the future design of simulation-based optimisation techniques. Currently, a Java tool is under development.

REFERENCES

Derrick, E. J., Balci, O., Nance, R.E., 1989. A Comparison of Selected Conceptual Frameworks for Simulation Modelling. *Proceedings of the 1989 Winter Simulation Conference*. 711-718, Blacksburg (VI, USA).

Fritz, D.G., Sargent, R.G., 1995. An Overview of Hierarchical Control Flow Graph Models. Proceedings of the 1995 Winter Simulation Conference. 1347-1355, Arlington (Virginia, USA).

Legato, P., Mazza, R.M., Trunfio, R., 2008. Simulation-based optimization for the quay crane scheduling problem. *Proceedings of the 2008 Winter Simulation Conference*. Miami (FL, USA).

Legato, P., Trunfio, R., 2007. A Simulation Modelling Paradigm for the Optimal Management of Logistics in Container Terminals. *Proceedings of the 21th European Conference on Modelling and Simulation*. 479-488 , Prague (Czech Republic).

Petri, C.A., 1962. Kommunikation mit Automaten, Ph.D. Thesis, Schriften des Institutes füt Instrumentelle Matematik, Bonn.

Pidd, M., 2004. *Computer simulation in management science* (5th edition). John Wiley & Sons: Chichester.

Pidd, M., Castro, R.B., 1998. Hierarchical Modular Modelling in Discrete Simulation. *Proceedings of the 1998 Winter Simulation Conference*. 383-389, Washington (DC, USA).

Sargent, R., 1997. Modelling queuing systems using hierarchical control flow graph models. *Mathematics and Computers in Simulation* 44(3): 233-249.

Schruben, L.W., 1983. Simulation Modelling With Event Graphs. *Communications of the ACM* 26(11):957-963.

AUTHORS BIOGRAPHY

PASQUALE LEGATO is an Assistant Professor of Operations Research at the Faculty of Engineering (University of Calabria). His home-page is http://www.deis.unical.it/legato.

DANIEL GULLÌ is devoted to research in numerical simulation at the Center for High-Performance Computing and Computational Engineering (CESIC) in NEC Italy.

ROBERTO TRUNFIO is currently pursuing a Ph.D. degree in Operations Research from the University of Calabria and is a logistics engineer at the CESIC in NEC Italy.

RUNWAY CAPACITY OPTIMIZATION: AIRCRAFT SEQUENCING IN MIXED MODE OPERATION

Olatunde Temitope Baruwa[a], Miquel Àngel Piera[b]

[a, b]Department of Telecommunications and System Engineering (LogiSim)
Universitat Autònoma de Barcelona, Barcelona, Spain.

[a]OlatundeTemitope.Baruwa@uab.cat, [b]MiquelAngel.Piera@uab.es

ABSTRACT

To meet future air transportation requirements, technology advancement has paved a way for the application of decision support systems to optimize strategic and tactical operations. This paper presents a simulation approach for the optimal sequencing of aircrafts on a single runway operating in a mixed mode (arrival and departure) and runway system capacity assessment using State Space Analysis. System dynamics are specified using the Coloured Petri Nets (CPN) formalism. The approach is capable of automating the decision activity (scheduling), analyzing different scenarios of scheduling policies and optimizing the runway capacity at any given time based on actual or dynamic traffic flow. It is aimed at validating not only the expected benefit of capacity and safety but also the benefits on efficiency from the air traffic controller's perspective.

Keywords: air traffic control, scheduling, coloured petri net, decision support system

1. INTRODUCTION

The increasing demand for air transportation services in recent times has called for the participation of the simulation community to provide and deploy solutions that would improve the current state of the industry. Specifically, efficient management of available infrastructure, for example, runways, taxiways, aprons etc in Air Traffic Management (ATM) has been a major concern in the industry. A challenging problem inherent to the air traffic flow management is how to maximize capacity given the available infrastructure in the face of growing demand.

The growth in air travel is outstripping the capacity of the airport and air traffic control (ATC) system, resulting in increasing congestion and delays. However, a misunderstanding of the poor utilization of the available infrastructure usually leads to greater investments in additional runways and extensive pavements for taxiways and aprons. In order to avoid this expensive approach, it is important to remove non-productive operations due to poor scheduling approaches. Thus, simulation models could help to analyze the operational efficiency of the current traffic control procedure and propose new viewpoints and decision support tools to address air traffic throughput.

Air traffic controllers have been able to maintain a safe and orderly flow of air traffic in a conservative manner. The use of traditional sequencing approaches mainly based on ICAO procedures (still using voice communications) hinders considerably the use of advanced decision support system (DSS). Though the traditional approach might seem to be a good one for scheduling landing and departure aircraft operations under low traffic conditions, this approach becomes inefficient during peak hours under a workload of a certain aircraft mix. A limiting feature of the sequencing pattern which in turn affects capacity is ATC regulation rules requiring a minimum safety separation between different types of aircrafts to avoid wake turbulence. Other factors influencing runway capacity include: air traffic control, characteristics of demand, environmental conditions and the layout and design of the runway system.

A number of different approaches have been employed by researchers for scheduling aircraft landings and departures in an effort to maximize runway capacity while minimizing delays. These approaches include: Queuing Models, Analytical Approaches and Computer Simulation. Bäuerle et al. (2007) presents a queuing model and a number of heuristic routing strategies to minimize the waiting time of arriving aircrafts (static) with one or two runways. Chandran and Balakrishnan (2007) use a dynamic programming algorithm to generate schedules of airport runway operations that are susceptible to perturbations. Several exact and heuristic optimization methods for scheduling arriving aircrafts and comparing these with integer programming formulations are given in Fahle et al. (2003). In Bolender et al. (2000), a number of scheduling strategies are analyzed in order to determine the most efficient means of scheduling aircraft when multiple runways are operational and the airport is operating at different utilization rates. Hu and Chen (2005) introduce the concept of receding horizon control (RHC) to the problem of arrival scheduling and sequencing in a dynamic environment. A multiple runway case of the static Aircraft Landing Problem is

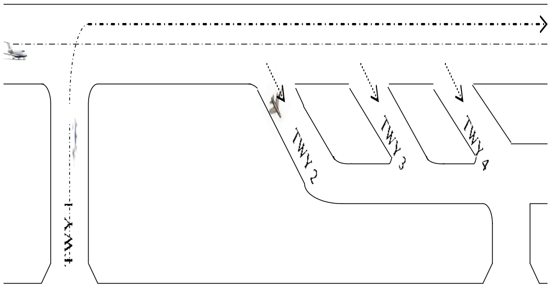


Figure 1: A Simple Runway

considered in Pinol and Beasley (2006), employing scatter search and population heuristic approaches.

Kovatcs et al (2005) developed a timed stochastic coloured petri net model of a single runway that analyzes the effect of the taxiway's availability on the runway capacity with respect to a given schedule. Some assumptions in this model cannot be adapted to practice.

Although airline business models help to understand the delay-capacity relationships, they do not provide accurate estimate of average delay except for simple situations, since they are used to evaluate best practices as a steady state solution when analyzing delay and throughput in the air traffic system. Not all models presented in the literature consider real aspects of aircraft landing and take off operations.

In order to accommodate future air traffic needs, a "paradigm shift" supported by the state-of-the-art and innovative technologies is required (SESAR, 2005 and its equivalent NGATS). The SESAR (Single European Sky ATM Research Programme) project, currently in the development phase is aimed at developing new generation air traffic management system capable of ensuring the safety and fluidity of air transport worldwide over the next 30 years. New communication requirements consist of a data link to automate the air traffic control system for scheduling aircrafts with less human intervention. Thus, some of these new technology advancements have paved a way for the application of simulation in the air transportation industry.

The primary objective of the research introduced in this paper is to develop a simulation model to automate and optimize the scheduling decision activity of air traffic controllers when prioritizing the next landing or departure operation while maintaining a high safety level (ie. wake turbulences), reducing costs and minimizing environmental hazards. Thus, an approach based on a simulation model that describes the system dynamics developed in CPN Tools is presented. Though simulation allows the modeller to visualize and understand how the system works, as a decision support tool it is only capable of reporting or exploring a small number of scenarios. The implemented approach is capable of automating the decision activity (scheduling), by analyzing different scenarios of scheduling policies and optimizing the runway capacity at any given time based on actual traffic flow. It is aimed at validating not only the expected benefit of capacity and safety but also the benefits on efficiency from the air traffic controllers' perspective.

The paper is organized as follows: Section 2 describes the air traffic flow problem and modes of operation; Section 3 presents the CPN model and the benefits obtained by using Coloured Petri Nets as a modelling formalism for logistic systems; Section 4 presents the scheduling strategies employed; Section 5 describes the results obtained while Section 6 gives a summary of the paper and ideas for future research.

2. THE PROBLEM

Given a set of aircraft fleet mix competing for landing or take off operation on a single runway, the optimization problem can be simplified to find an automatic optimal scheduling policy and to allocate landing or take off times during heavy traffic flow. The objective is to maximize the runway capacity such that: only one aircraft can occupy the runway at a time, the minimum separation requirement between aircrafts is met, and the precedence of arrivals over departures. Other objective that will be considered is: minimizing delays whilst in air and on land in terms of fuel consumption and environmental hazards. Figure 1 shows the layout of a simple runway.

The aircraft fleet mix can be modelled as a three weight classes based on the maximum take off weight capacity (Chandran B. and Hamsa B., 2007). They are classified as: heavy (H), medium (M) and light (L). The minimum time-based separation requirement matrix between the different classes of aircrafts using the same runway as presented in Martinez J.C et al. (2001) is shown in table 1. The separation time ensures that the runway will be free when a trailing aircraft is scheduled to touch down or enter the runway. This is a measure to avoid collisions and wake turbulence in air and on the runway. With this, only one aircraft would be able to occupy the runway at a particular point in time. This goes a long way in reducing runway capacity. Another striking feature that influences runway capacity is the number of available exit taxiways for aircrafts. It brings about variation in runway occupancy time (ROT) for the different aircraft classes. The average ROT and touch down time for the aircraft categories according to the approach speed is given in table 2 (Martinez J.C et al., 2001). In practice, arrivals are generally given absolute priority over departures. Departures are released when suitable gaps occur in the arrival stream. The runway is said to be under-utilized if it is operating in a segregated mode (Ashford N. 1992). Runway capacity can be substantially increased with mixed operations.

During arrival or departure, a controller directs each aircraft. Upon approaching an airport at which a landing is to be made, the pilot is required to make contact with a controller so that separation of all aircrafts can be provided. If the path is clear, the controller directs the pilot to the runway; if the airport is busy, the aircraft is fitted into a traffic pattern with other aircraft waiting to land - a holding area away from the runway called Terminal Control Area (or Terminal Manoeuvring Area, TMA). This is an area where the aircrafts hold until the control units are ready to position them into an approach sequence to land. When the runway becomes available, the waiting aircraft is directed to the Instrument Landing System (ILS). Several aircrafts can be on the ILS at the same time, several miles apart. The aerodrome controller directs the plane to the proper runway and then informs the pilot about conditions at the airport, such as weather, speed and direction of wind, and visibility. The procedure is reversed for departures. The controller directs the aircraft waiting on taxiway for instructions to enter the runway and then informs the pilot about conditions at the airport. The controller also issues runway clearance for the pilot to take off. Once in the air, the aircraft is guided out of the airport's airspace by the controller.

Table 1: Minimum Time between Successive Arrivals and Departures (seconds)

Leading Plane	Trailing Plane					
	Heavy		Medium		Light	
Heavy	96	60	120	90	144	120
Medium	72	60	72	60	96	90
Light	72	60	72	60	72	60

Table 2: Runway Occupancy and Touch Down Times

Aircraft Type	Runway Occupancy Time		Touch Down Time
	Landing	Departure	Landing
Heavy	55	38	60
Medium	50	43	65
Light	45	50	70

3. THE CPN MODEL

CPNs are well known for their capability in simulating and analyzing discrete-event system (Jensen K. 1997). In this section a CPN model is illustrated to determine the scheduling strategies, expected landing and take-off time, runway capacity assessment, total fuel consumption and air quality factor index.

CPN has been chosen as the modelling formalism due to its ability to describe the complete structure of a system together with its behaviour and the information about the system state (Narciso M. and Piera M.A, 2001) through the use of a functional programming language. PN is a bipartite directed graph describing the structure of a discrete event system, while the dynamics of the system is described by the execution of the PN. A PN is coloured if the tokens are distinguishable. The main CPN components are: state vectors, arc expressions and guards, colour sets, places and transitions. See [(Jensen K. 1997), (Narciso M. and Piera M.A, 2001)] for the description of these terms and tutorial on CPN.

To model the air traffic flow operations as a discrete event system, it is necessary to define events that are relevant. CPN allows the representation of a system in a compact structure with few places and transitions. The model is implemented in CPN Tools software developed and maintained by the CPN Group, University of Aarhus, Denmark for validation and verification purposes. It is then transferred to another CPN simulator tool developed at the Universitat Autonoma de Barcelona, Spain for evaluating the state space (See section 4). Figures 2 and 3 shows the CPN model of arrivals and departures respectively. The model consists of 13 place nodes and 5 transitions (T1, T2, T3, T4 and T5) that describe the system dynamics. The meanings are given in table 4, 5 and 6.

The model has 2 parts; one for the arrivals and the other for departures, competing for the shared resource (Runway). The information enclosed in TMA node consists of 4-tuples (aircraft type, average runway occupancy time, fuel consumption rate per second and time) describing aircrafts waiting in the TMA to be positioned at the ILS for landing. Place node TW represents the aircrafts waiting to take off. Place node S contains the matrix of minimum separation time for landing and take off. Other measures used to determine delays are: total fuel consumption rate and a weighted air quality index. The fuel consumption rate measured in litres/secs is used to estimate the total delay for aircrafts in holding trajectory while the weighted air quality index is a penalty for departing aircrafts' delay. The values are given in table 3.

Transition T1 is an event that positions arriving aircrafts from the TMA on the ILS while maintaining the minimum separation time. The sequence in which the aircrafts are placed in the ILS is independent of the aircraft type. Transition T4 places the aircrafts waiting to depart at the apron for take off while Transitions T2, T3, T5 are events describing touch down, exit from the runway and aircraft take off respectively. The time is kept at zero for aircrafts in the TMA and taxiway. This is to measure the effect of continuous demand on the runway capacity during heavy traffic flow.

Table 3: Fuel Consumption and Weighted Air Quality Index

Aircraft Type	Fuel Consumption (/secs)	Air Quality Index
	Landing	Departure
Heavy	4	60
Medium	2	40
Light	1	20

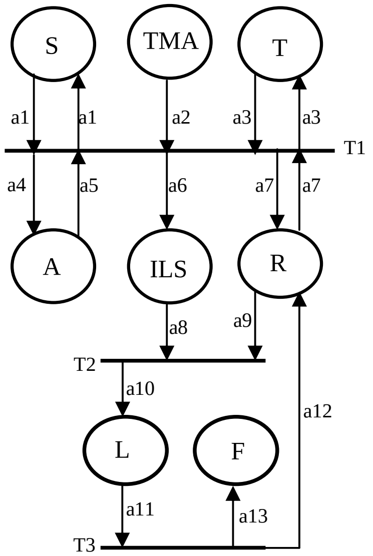


Figure 2: CPN Model of Arriving Aircrafts

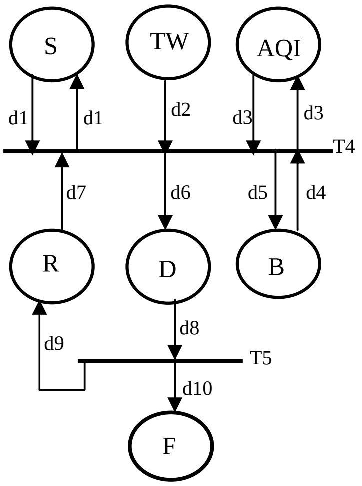


Figure 3: CPN Model of Departing Aircrafts

Table 4: Colour Description

Colour	Definition	Colour Description
X,Y	int 1..3	Aircraft type, Leading aircraft
L	int	Landing/take off occupancy time
Lt	int	Computed touch down time for scheduled landing aircrafts
C	int 1..4	Fuel consumption (litres/secs)
T, S	int	Time at the TMA/Taxiway, Separation time
T1, T2	int	Cumulative separation times, Runway utilization time
To, Ts	int	Cumulative occupancy time, Estimated scheduled time
T3, Tt	int	Time at touch down, Touch down time for aircrafts on the ILS
O, I	int	Operation identifier; landing or departure, integer
U, Q	int	Cumulative fuel consumption, Cumulative air quality index
Sm	Product Y, X, S	Safety Matrix
Tm	Product X, L, C, T	Landing aircrafts information waiting in the TMA
Td	Product X, Tt	Touch down information
Sc	Product Y, T1	Cumulative separation
Is	Product X, L, C, Lt, Tt	Aircrafts on the ILS
Ri	Product I, T2, U, Q, I, To	Runway information
La	Product X, L, T3, Tt, U, Q, I, To	Landing aircrafts occupying the runway
Tw	Product X, T	Departing aircraft information waiting on the taxiway
Aq	Product X, L, Q	Air quality information
Da	Product X, L, Ts, Q, U, To	Departing aircrafts occupying the runway
Fs	Product X, O, Ts	Scheduled aircrafts

Table 5: Colour Petri Net Place Description

Place	Colour	Description
A, B	Sc	Cumulative separation for arrival and departure respectively
AQI	Aq	Represents air quality information
D	Da	Departing aircrafts occupying the runway
F	Fs	Scheduled aircrafts – final state
ILS	Is	Represents aircrafts positioned on the ILS
L	La	Landing aircrafts occupying the runway
R	Ri	Runway information
S	Sm	Represents safety matrix
T	Td	Describes the touch down time for arriving aircrafts
TMA	Tm	Represents arriving aircrafts in holding trajectory
TW	Tw	Represents departing aircrafts waiting on the taxiway

Table 6: Arc Expression

Arc	Expression
a1, a2	1’(y, x, s), 1’(x, l, c, t)
a3, a4	1’(x, tt), 1’(x, M(tt, t + s, t2 - tt))
a5, a6	1’(y, t1), 1’(x, l, c, M(tt, t + s, t2 - tt) + tt, tt)
a7, a8	1’(1, t2, u, q, 0, to), 1’(x, l, c, lt, tt)
a9	1’(1, t2, u, q, 0, to)
a10	if (lt >= t2) then 1’(x, l, lt, tt, u + c * (lt - tt), q, 0, to + l) else 1’(x, l, t2, tt, u + c * (lt - tt), q, 1, to + l)
a11, a12	1’(x, l, t3, tt, u, q, i, to), 1’(1, l + t3, u, q, i, to)
a13	1’(x, 1, t3 - tt)
d1, d2	1’(y, x, s), 1’(x ,t)
d3, d4	1’(x, l, q1), 1'(y, t1)
d5, d6	1’(x, M(tt, t + s, t2 - tt)), 1’(x, l, M(t, t2, t1 + s), q + l * tt, u, to + q1)
d7, d8	1’(1, t2, u, q, 0, to), 1’(x, l, ts, q, u, to)
d9, d10	1’(1, ts + l, u, q, 0, to), 1’(x, 2, ts)
M	fun M(x ,y, z: INT) = (if x >= y andalso x >=z then x else if y >= x andalso y >= z then y else z);

Table 7: Scheduling Solution

Place Node	R	F
Final State	1'(1,1443,6936,0,0,940)	1'(1,1,192) + 1'(1,1,96) + 1'(1,2,1405) + 1'(1,2,60) + 1'(2,1,312) + 1'(2,1,384) + 1'(2,1,456) + 1'(2,2,1075) + 1'(2,2,1285) + 1'(2,2,1345) + 1'(3,1,552) + 1'(3,1,624) + 1'(3,1,696) + 1'(3,1,768) + 1'(3,1,840) + 1'(3,2,1015) + 1'(3,2,1165) + 1'(3,2,1225) + 1'(3,2,571) + 1'(3,2,955)

Key: F - (aircraft type, operation identifier, scheduled time)
Aircraft type: 1 – light, 2 – medium, 3 – heavy
Operation identifier: 1 – landing, 2 – departure

4. SCHEDULING STRATEGY

State Space analysis permits the evaluation of a wide range of options leading to better decision making: it permits the comparison of various alternatives depending on the actors involved; it allows timely policy and real time decisions to be made. However, the amount of nodes generated can grow to computationally prohibited size when applied to real systems. The CPN simulator tool is used to generate and explore the coverability tree.

Each time a new state is generated, the markings are checked against the previous states on the same path. If the new marking has been generated previously, it is labelled as “old”. The tool will not explore enabled transitions associated with this new state. However, a new state is labelled as “dead end” if there is no enabled transition. The tree is further explored if the same state has not been generated until the final state is established. In addition, a new state is not generated for a node where there is a successive increment of tokens. A symbol “ω” is introduced to stop the further expansion of the path. A simple example of the CT for a Petri net is presented in figure 4.

The underlying idea is to transform a scheduling problem to a search problem, that is, to obtain a path from a certain system state to a desired goal state in a tree structure that represents the problem state. CPN formalism provides an easy way to introduce new restrictions to reduce the size of the state space under acceptable computational time: restricting the search space by eliminating some possibilities that will not lead to a feasible solution and specification of constraints on events firing. The tool allows the modeller to specify the final state required or desired to be reached.

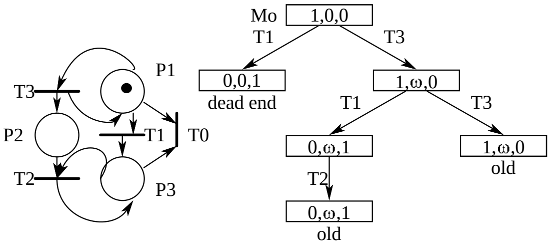


Figure 4: PN Coverability Tree Example

Though the state space is used to determine the various scheduling strategies, a number of other strategies that supports the state space to drive the system to optimal are implemented in the model. These include:

1. Runway utilization is kept open for arriving and departing aircrafts. This is to allow the state space explore all the possible scheduling policies that are obtainable for aircrafts waiting in queue. Several possibilities exist: all arriving aircrafts can be scheduled before the departing ones; arrivals can be scheduled after departures, two departures can be scheduled to take off after every arrival and vice versa.
2. An aircraft is scheduled to land if and only if the runway will be free at touch down and the minimum separation time between the leading and trailing aircraft is adhered to.
3. A departing aircraft is authorized to take off if the ROT is less than or equal to the time a trailing aircraft will touch down. This is to avoid collisions and wake turbulence on the runway.
4. The further expansion of nodes on a path is disabled if the touch down time or a take off time of a trailing aircraft is greater than the time a leading aircraft will leave the runway. This is a restriction to eliminate the search for infeasible solutions as wake turbulence and land side accidents are bound to occur.

A scheduling policy is considered optimal if it has the best sequence of events that maximizes the runway capacity while fulfilling the security and minimum delay requirements. The model allows the formulation of different objective functions according to the preference of the actors involved. For instance, an airline may be more interested in the scheduling policy that minimizes the total fuel consumption of her aircraft whilst in queue.

5. RESULTS

The model is driven by an aircraft mix index of 50% H; 30% M; 20% L; for both landing and departure. The aircraft mix can be easily modified to reflect any other mix according to aircraft flows at any point in time.

The preferred measure is the saturation runway capacity; the maximum number of aircraft that can be handled during a given period under conditions of continuous demand, expressed in operations (i.e. arrivals, departures) per hour.

The scheduling policy that leads to a feasible solution with the estimated landing and take off time is given in table 7. The total time for the operation is given as 1443 seconds while the total fuel consumption is 6936 litres.

The maximum runway capacity under this mix is evaluated as 50 operations per hour (20/1443*3600) and the runway will be occupied two-thirds of an hour (50*940/20 = 2350 seconds). The results presented validate the capability of the runway for handling aircraft flows when operating in mixed mode.

6. CONCLUSIONS

Runway capacity analysis is undertaken for two purposes; to measure the capability of the runway for handling aircraft flows and continuous demand and to estimate delays experienced in the system at different levels of demand. The study has dealt with the two objectives with delays measured by fuel consumption of arriving aircrafts in air and air quality index of departing aircrafts on the land side.

Further research work will focus on simultaneously allocating taxiways to each aircraft with the scheduled time and rescheduling or re-feeding landed aircrafts into the system for subsequent departure. With this, the runway capacity can be improved since aircraft occupancy time differs according to the aircraft type and allocated runway.

REFERENCES

Ashford N. and Wright P.H, 1992. *Airport Engineering*. New York; John Wiley & Sons Inc

Bäuerle N., Engelhardt-Funke O., and Kolonko M., 2007. *On the waiting time of arriving aircrafts and the capacity of airports with one or two runways*. European Journal of Operational Research, Vol. 177, Issue 2, pp 1180-1196.

Bolender M., and Slater G., 2000. *Evaluation of Scheduling Methods for Multiple Runways*. Journal of Aircraft, Vol.37 No.3, pp 410-416.

Chandran B., and Balakrishnan H., 2007. *A Dynamic Programming Algorithm for Robust Runway Scheduling*. Proceedings of the 2007 American Control Conference, July 11-13, 2007, New York City (USA).

Fahle T., Feldmann R., Götz S., Grothklags S., and Monien B. 2003. *Computer Science in Perspective - The Aircraft Sequencing Problem*. Springer Berlin / Heidelberg, pp 152 – 166.

Hu X., and Chen W., 2005. *Receding Horizon Control for Aircraft Arrival Sequencing and Scheduling*. IEEE Transactions on Intelligent Transportation Systems, Vol. 6, No. 2.

Jensen K., 2003. *CPN Tools, University of Aarhus, Denmark*. Available from: http://www.daimi.au.dk/CPNTools/

Jensen K., 1997. *Coloured Petri Nets: Basics Concepts, Analysis Methods and Practical Use*. Springer, Vol. 1, 2, 3.

Kovatcs A., Nemeth E., and Hangos K., 2005. *Modelling and Optimization of Runway Traffic Flow using Coloured Petri Nets*. Intemational Conference on Control and Automation (ICCA2005), June 27-29, 2005, Budapest (Hungary).

Martinez, J.C, Trani, A.T, Ioannou, P.G. 2001. *Modeling Air-side Airport Operations using General-Purpose Activity-based Discrete-Event Simulation.* Journal of the Transportation Research Board 1744, pp. 65-71.

Narciso M.E and Piera M.A. 2001, *Coloured Petri Net Simulator: A Generic Tool for Production Planning.* ETFA'2001 - 8th IEEE International Conference on Emerging Technologies and Factory Automation, Antibes, 2001.

Pinol H and Beasley J.E., 2006. *Scatter Search and Bionomic Algorithms for the Aircraft Landing Problem*. European Journal of Operational Research, 171, pp 439–462.

http://www.sesar-consortium.aero/

AUTHORS BIOGRAPHY

OLATUNDE T. BARUWA holds a bachelor’s degree in Electronic and Computer Engineering (Lagos State University, Nigeria, 2004) and a Master’s degree in Management (Universitat Pompeu Fabra, Spain, 2007). He is currently a PhD student in the Department of Telecommunication and Systems Engineering at the Universitat Autònoma de Barcelona, Spain where he is developing his dissertation and a member of LogiSim, a Modelling and Simulation Institution sponsored and founded by the local government of Catalunya. Prior to studying at UPF, he worked at the Lagos Business School, Nigeria for 2 years. His research interests are in the areas of modelling, simulation and optimization of logistic systems, industrial processes and supply chain management.

MIQUEL ÀNGEL PIERA I EROLES (DR.) received his MSc (Control Engineering) from the University of Manchester Institute of Technology in 1990 and his PhD degree from the Autonomous University of Barcelona (Spain) in 1994. He participates in industrial research projects in the logistics and manufacturing field and at present he is Co-director of LogiSim, a Modelling and Simulation Institution sponsored and founded by the local government of Catalunya. Professor Piera has been the coordinator of the Spanish Simulation group of the International Federation of Automatic Control. Recently, he has published a modelling and simulation book that is being used for teaching in many Spanish Universities.

GREAT BUT FLAWED EXPECTATIONS: ON THE IMPORTANCE OF PRESUMPTIONS AND ASTONISHMENT IN MODEL AND SIMULATION BASED RISK MANAGEMENT

Marko Hofmann[(a)], **Thomas Krieger**[(b)]

Institut für Technik Intelligenter Systeme (ITIS e.V.)
Universität der Bundeswehr München
Werner-Heisenberg-Weg 39
Germany - 85577 Neubiberg

marko.hofmann@unibw.de[(a)], thomas.krieger@unibw.de[(b)]

ABSTRACT

Unexpected detrimental events probably pose the most dangerous threat to every planning activity. They are the consequence of both explicit and unconscious presumptions made during the planning process. These presumptions are the manifestation of the modeler's own expectations, which can be seriously flawed. Model and simulation based risk management tries to identify potentially dangerous presumptions (for the real world planning) by looking for astonishing results in models in general and simulations in particular. The astonishment is triggered by (simulation) events that are violations of model assumptions (the model specific instantiations of the presumptions) or events which are simply counter-intuitive. The main idea of this approach is illustrated using examples taken from reliability theory. This choice has been made for didactical purposes: the analytical perspicuity of these examples is much better than the one of complex simulation models. Subsequently, the benefits of the approach are demonstrated for military conflict simulation models.

Keywords: uncertainty, simulation supported risk management, modeling assumptions, analysis and exploration, reliability theory

1. INTRODUCTION: MANAGING RISK

Risk management has become a paramount task of modeling and simulation in a great variety of applications. Hitherto finance markets and military endeavors have been the most prominent domains of risk management, but it seems to be indispensable for industrial applications, too. From a generalized point of view, risk (in the broader sense) has two dimensions: risk in the narrower sense and real uncertainty (Knight 1921). Risk (in the narrower sense) is associated with known dangerous events and the possibilities of their occurrence. These possibilities are regarded to be assessable, using "hard facts" (e.g. frequencies) (Risk type 1). Real uncertainty is an attribute of known dangerous events for which the possibility of occurrence is indeterminable on objective grounds (Type 2) and of completely unexpected events, which reveal their detriment only after they have happened (Type 3).

The more modern classification of parametric and structural uncertainty (which can not be attributed to a single origin) has a slightly different meaning, but addresses approximately the same distinction as type 2 and 3 uncertainty. Parametric uncertainty means that we know the relevant factors for a given phenomenon, but miss the exact (initial) values of these factors. In other words, we have good empirical evidence that the causal reasoning of the model we use is an adequate representation of the relations in the real world. What is sometimes hard to find are the "right" parameters for the model. Parametric uncertainty is roughly equivalent to uncertainty type 2 in the modified Knight's classification, but it also includes type 1, if we see probabilities as special expressions of parameter uncertainty. Structural uncertainty means that we are not sure if we know all the relevant factors and that we most probably do not know their interdependencies. Or, in other words, there are serious reasons to believe that the model we use to represent the phenomenon is at least incomplete. Hence, structural uncertainty ideally reflects the concept of type 3 uncertainty in the former classification, only substituting unknown factors for unknown events.

Ex ante, all three types of risk are equally important. In practice, on the contrary, the amount of work dealing with the first type of risk dominates the other two. This disequilibrium is due to the human inclination to operate on mathematically treatable information. Type 2 events are therefore often transformed into type 1 events using subject matter experts and their estimations as substitutes for objectively generated probabilties. Slightly simplifying matters, this paper treats risk type 2 as parametric uncertainty and risk type 3 as structural uncertainty.

The remainder of this paper is organized as follows: Section 2 outlines the difference we make between planning presumptions and modeling assumptions. Section 3 introduces the concept of astonishment. Section 4 provides the reader with an example of an astonishing result in the field of parametric uncertainty. Section 5 gives a second example which can be attributed to structural uncertainty. Section 6 discusses tactical wargaming, in general. Section 7 presents a simulation based wargaming method for parametric uncertainty, section 8 repeats it for structural uncertainty. Section 9 concludes the paper by reiterating its contributions and suggests some conclusions and future research directions.

2. PLANNING PRESUMPTIONS – MODELING ASSUMPTIONS

Making plans for the future of complex social systems is always affected by personal presumptions. They are unavoidable for many reasons. First, the perception and conception of human beings are limited by their experience and cognitive constraints (bounded rationality, see, e.g., Gigerenzer and Selten 2002). Second, planning in social systems is impossible without making predictions on the behavior of other humans. These predictions are necessarily unreliable.

Examples for such presumptions are:

- the equivalence of political “solutions” in different cultures,
- the superiority of western thinking and governing to all other forms,
- the impossibility of anomalies in hitherto well-understood systems,
- taking statistical correlations as causal dependencies.

In general, only a small part of such presumptions is mentioned during the planning process, a much greater part is unconscious, but detectable via questioning. Some presumptions are even difficult to detect, because they are deeply hidden in attitudes and beliefs. Such presumptions are classical examples of risk type 3.

If we deliberately reflect on our plans and make planning support models, the conscious part of these presumptions become modeling assumptions.

Modeling is seen here as a process that, if successful, helps a subject S to solve a problem P situated in an original (system) O at a given time T. The model M of the original O is always regarded as an abstraction (Stachowiak 1973). The notion of assumptions is concentrated in the following definition: An assumption A is a hitherto not (or insufficiently) empirically corroborated statement (in the sense of an assertion)

- of a subject S,
- about an original O,
- with the intention (purpose) I relative to
    - a problem P situated in O and
    - a timeframe T.

Some typical assumptions used in many (simulation) models are:

- Stability of processes,
- Uniformity of interactions over time and space,
- Linearity of interrelations,
- Processes have reached equilibrium,
- Empirical data fit approximately uniform, exponential, normal etc. distribution,
- Independence of statistical parameters.

In the narrow sense used here (for a broader view see Hofmann 2003), model assumptions are always explicit and therefore conscious, whereas some planning presumptions can still remain unconscious even after the modeling.

Model assumptions are examples for uncertainty of type 2.

In the following sections and examples, this distinction between explicit modeling assumptions as instantiations of risk type 2 and hidden personal presumptions as examples of risk type 3 will be further clarified.

3. ASTONISHMENT

With regard to the two types of risk scrutinized here, astonishment can be triggered by the violation of an explicit assumption or a hidden presumption. In the first case, during the realization phase an event occurs that seemed highly improbable during the planning phase. Consequently, the contrary had been assumed. In the second case, an event occurs which has not been expected during the planning. Some of such events can be advantageous for the own goals, but, in general, they are detrimental, because when things turn out different as supposed to be, plans cannot be accomplished optimally. Planners and decision makers must therefore try to reduce the amount of astonishing events during the realization phase.

The methodology to cope with events that have been considered improbable (type 2) differs from that applicable to unexpected events (type 3). In the first case (parametric uncertainty) it is always possible to reflect on the assumptions made. In tactical wargaming, a method developed for the German army, for example, all explicit modeling assumptions are deliberately taken as violated. The criticality and plausibility of this violation is estimated and contingency plans can be developed (see section 7).

Since we simply do not know unexpected events we cannot use this approach for type 3 risk (structural uncertainty). The default strategy of military decision makers for type 3 risk, for example, is the creation of reserves. Nevertheless, it would be advantageous to find as many potentially detrimental events in advance, because the planning of countermeasures could be much more specific (see section 8). The basic idea is to generate some astonishing results/events in a simulation, thereby reducing the amount of astonishing results/events in reality.

Before we discuss the application of these concepts within the framework of wargaming based on complex military simulation models, we will highlight the core idea with two analytical examples presented in the next two sections. These examples are taken for their simplicity, clearness and mathematical accuracy. Although the hard probabilistic reasoning might be somewhat misleading, the examples focus on the essential. The first example will demonstrate an unexpected effect in parameter variation and the second example is intended to illustrate a structural surprise.

The main difference between these examples and risk management in real world applications, like wargaming, is, that the astonishing results in the examples can be proven to be hard facts, whereas the simulation results in wargaming are generally only conjectures.

4. ON THE SAFETY OF AIRCRAFT WITH TWO OR FOUR ENGINES

In this section we consider an airline which wants to buy a new aircraft and has to decide between two types. One aircraft type (A_2) has two engines and the other type (A_4) four engines. It is assumed, that the decision of the airline's manager which aircraft will be bought, is made on the sole basis of the reliability of the aircraft. That also implies, for instance, that the different numbers of passengers which can be carried, are not taken into account.

The airline defines with respect to own experiences that aircraft A_2 (A_4) is working (i.e., is still flying and can touch down safe), if and only if at least one (at least two) of the engines are still working. Furthermore, it is assumed that

1. the engines are working independently of each other (an assumption which may not be fulfilled in all practical situations), and
2. the probability p, that an engine is working is the same for both aircraft and all their engines (one might think of one type of engine which is used in both aircraft).

In the past the airline has already bought a lot of aircraft. For making their decision they use a simulation which – for a given value of p – computes the reliability of the aircraft. It is important, that in the past the decision were made on basis of simulations for values of p close to 1, because the engines itself are very reliable under good flight conditions. Also, a changing of p during the flight was not taken into account. Now, in recent years it turned out that this assumptions have been too optimistic. In extremely bad weather the reliability of the engines can significantly decrease. Due to fierce competition on the air transportation market airlines have to fly even under such extreme conditions.

Let $P_2(p)$ resp. $P_4(p)$ be the reliability of aircraft A_2 resp. A_4 (which only depends on p), i.e., the probability that the aircraft is still flying and can touch down safe. For values of p close to 1 the M&S-Team of the research group had observed in the past the relation $P_2(p) < P_4(p)$ holds for all p close to 1. They now make – most probably unconscious - the implicit assumption – coming from their experience - that $P_2(p) < P_4(p)$ for all $p \in [0,1]$. This would lead to an absolute preference of aircraft A_4.

Skeptical about this reasoning the airline's manager demands critical rethinking of all assessments made on reliability. Using their simulation system the research group starts with p = 1 and obtain $P_2(1) = P_4(1)=1$. This is an obvious result, since all engines are working with probability 1 and therefore – per definition – the aircraft is working. Taking p = 0.9, an already low reliability rate, they get $P_2(0.9) = 0.99$ and $P_4(0.9) = 0.9963$, which satisfies their expectation that $P_2(p) < P_4(p)$ for all $p \in [0,1]$. Assured in their opinion about the superior reliability of four-engine aircraft, they try p = 0.6 and obtain $P_2(0.6) = 0.84$ and $P_4(0.6) = 0.8208$, i.e., $P_2(0.6) > P_4(0.6)$, and are absolutely astonished. This should be impossible! It seems unbelievable to them, that an aircraft with four engines can be less safe than a two-engine aircraft.

In fact, this result is not surprising at all. Using elementary probability calculus we get (see, e.g., Rohatgi 1976)

$$P_2(p) = 1 - (1 - p)^2$$

and

$$P_4(p) = 1 - (1 - p)^4 - 4\,p\,(1 - p)^3.$$

Both functions are presented graphically in Figure 1.

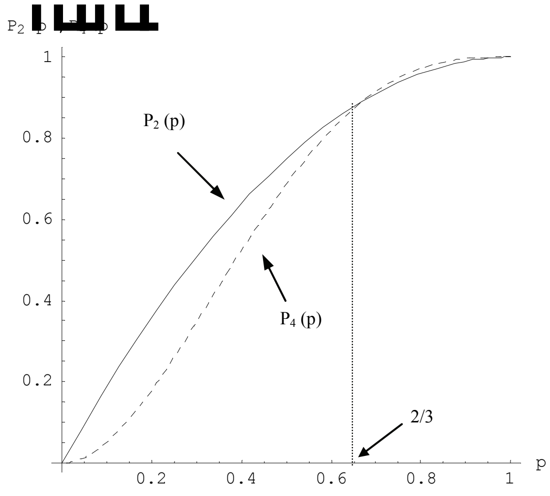


Figure 1: The Reliability Functions of Aircraft with two resp. four Engines.

We see that in case of $p \in (2/3,1]$ A_4 is more reliable than A_2, while in case of $p \in [0,2/3)$ A_2 is indeed more reliable than A_4. Consequently, the expectation $P_2(p) < P_4(p)$ for all $p \in [0,1]$ is obviously false. The maximum difference between the system reliabilities P_2 and P_4 are 0.0128917 for $p \in (2/3,1]$ and 0.179558 for $p \in [0,2/3)$. That implies that a two-engine aircraft is, at worst, only 0.98 times less safe than a four-engine aircraft under good conditions but a four-engine aircraft can be as far as 0.52 times less safe than a two-engine aircraft under extremely critical conditions.

Of course, this reasoning cannot be a decisive argument to buy (or construct or use) only two-engine aircraft, since it completely neglects the actual distribution of p in reality (which may even be hard to find). However, the research group should have realized, that some assumptions they have made in judging the reliability of aircraft, may not hold in all cases. Such a self-critical attitude is the cornerstone of risk management in real applications.

By the way, in a group of 13 computer scientist and mathematicians we asked to estimate the two reliability functions, only one person has made the right guess.

5. NON-MONOTONIC RELIABILITY FUNCTIONS

A company wants to set up a network with three nodes and three edges (see Figure 2).

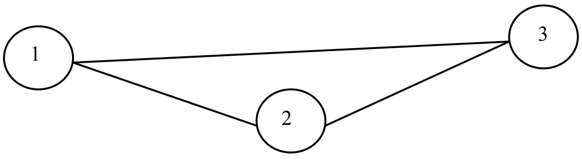


Figure 2: The Network.

It is known that

1. The nodes are working independent of each other with probability p.
2. The edges are faultless independent of each other with probability q.
3. Nodes and edges are working independent of each other.

The company defines that the network is working (i.e., reliable) if and only if there is a connection between all faultless nodes. Furthermore it is defined that if all nodes are failed the network is supposed to be not working. Let P(q,p) be the probability that the network is working. The company´s M&S-Experts wants to get an idea of the reliability of the whole system depending on q and p. Due to their experience they expect that for a given q the reliability of the system is a monotone-increasing function of p. Moreover, they do not know any system that exhibits non-monotonic reliability behavior with respect to its components reliability. Furthermore they expect P(q,1)=1, since their first guess is, that if all nodes are failing then the whole system is failing and if all nodes are working with probability 1 the whole system should work with probability 1. Please, stop reading for a few seconds: What is your opinion? Which behavior of the reliability function P(q,p) are you expecting?

The M&S-Experts of the company uses a simulation in order to get insights into the behavior of the system. In case of q = 0.5 they obtain P(0.5,0.45) = 0.621 and P(0.5,1) = 0.5, i.e., the reliability of this system is in case of p = 0.5 larger than in case of p = 1. This outcome seems to them quite dubious. It is against the common understanding and the usual properties of reliability functions (see, e.g., Barlow and Proschan 1975). As mentioned above the experts are expecting a monotone-increasing function with P(0.5,1) = 1, which obviously is not fulfilled. These facts contradict their expert knowledge and this makes the system even more interesting to them.

The reliability function of the system can be computed as follows:

$$\begin{aligned} P(q,p) = {} & (1-q)^3\, 3\,(1-p)^2\, p \\ & + 3\,(1-q)^2\, q\,(3\,(1-p)^2\, p + (1-p)\, p^2) \\ & + 3\,(1-q)\, q^2\,(3\,(1-p)^2\, p + 2\,(1-p)\, p^2 + p^3) \\ & + q^3\,(3\,(1-p)^2\, p + 3\,(1-p)\, p^2 + p^3). \end{aligned}$$

The function P(q,p) is drawn in Figure 3 for the cases q = 0, 0.5, 1.

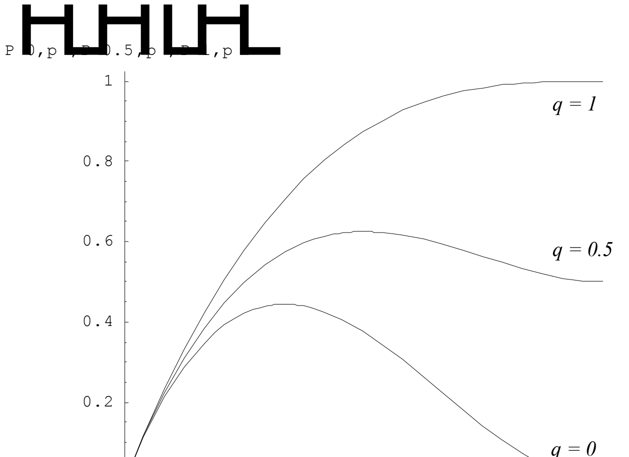


Figure 3: The Reliability Functions of the Network for the Cases q = 0, 0.5, 1.

As we can see we have P(0.5,0.5) > P(0.5,1). The reliability function P(q,p) – not only for fixed values of q - of the system as a function of q and p is depicted in Figure 4.

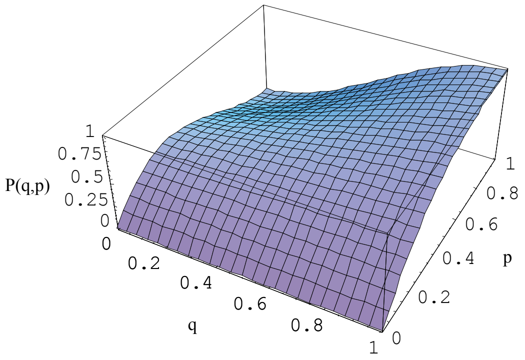


Figure 4: The Reliability Function P(q,p) of the Entire System.

The reason why this system does not behave like the experts expected it to do, is that it is not a monotonic system, i.e., if a failed element is repaired, then the system may change from a working system to a failed system. The reliability analysis of such systems is, in our opinion, a very interesting challenge for the future.

6. WARGAMING – A MILITARY TOOL OF RSIK MANAGEMENT

"Wargaming is a flexible instrument designed to develop, compare, and improve courses of action (COA)". This is a definition given by the German General Staff College. It can also be seen as one of the most important methods of risk management within the military domain. The origin of institutionalized Wargaming dates back to emerging Prussia and its General Staff in 19^{th} century and can be described as means to interactively play the *uncertain* development of a military (and later on non-military as well) operations. Both, Clausewitz and Moltke the elder saw the potential of Wargaming in a staff being better prepared for the incalculable course of an operation (v. Clausewitz 1832). They also advised it for the "free play" of its members' creativity and the intensified examination of planned COA (Hofmann and Lehmann 2007).

Military leadership is a domain where personal presumptions and, consequently, model assumptions have always played and are continuing to play a decisive role. The "model" used from commanders and their staffs is usually an abstract representation of own and enemy units on a terrain representation. Most of the explicitly mentioned presumptions of commander and staff are depicted within this environment. They are therefore visible or visualizable model assumptions (like the currently perceived situation or the supposed enemy commander's intent), which can be directly addressed. Some presumptions (like, just for example, the general superiority of mission-type tactics over order-type tactics and the higher value of special forces even in standard combat), in contrast, are not mentioned and therefore also not challenged.

Wargaming incorporates, even traditionally, two rather different aspects of investigation: analysis and exploration. In general, wargaming for COA *analysis* is based on a sequential process of some largely independent cycles (repetitions) of "moves" from at least two different teams within a gaming environment (sand table, map or computer based system). Within this paradigm it follows an Action – Reaction – Counteraction pattern using an impartial umpire to judge the outcome. The outcome can also be calculated using fixed rules (like in chess) or simulations (see section 7).

The central rationale of this analytical aspect of wargaming is the scrutiny of a succession of events that constitute a COA. Since all COA are given before the wargaming starts, the range of possible events is limited to what the team members regard as possible within that COA. From a generalized point of view, this kind of wargaming tries to reduce the uncertainty of known events and is therefore dealing with a special kind of parametric uncertainty (the decision relevant parameters of the COA). This technique can be applied on explicit model assumptions, too. One simply negates them and scrutinizes the effects.

The explorative aspect of wargaming is often hidden to the external observer, since it follows no general methodological rules and is basically a mental activity. Exploration means to think outside the standard analytical evaluation scheme. It is a questioning of own beliefs and assumptions and a speculative search for chances and exceptional risks. In such, explorative wargaming is an attempt to deal with structural uncertainty.

Both aspects of wargaming can be supported by combat or other military simulation systems. However, there is a significant distinction in the methodologies used to deal with parametric (already in use) and structural (currently tested within the German army) uncertainty (Hofmann and Junge 2008).

7. SIMULATION BASED WARGAMING FOR RISK TYPE 2 (PARAMETRIC UNCERTAINTY)

It is possible to use a stochastic simulation system instead of an umpire to evaluate the outcome of different COA. The wargaming factions implement their plan (COA) by fine-tuning the deployment of their units and giving initial orders. After the simulation has started all elementary combat processes, including off-road combat mode movement, reconnaissance and mutual attrition, are automatically simulated using random generators to mimic the effect of parametric uncertainty. In an interactive simulation it is possible and necessary to command the units by interrupting the simulation run, whereas in a closed simulation no human intervention is possible after the start.

The major advantage of interactive simulations is more flexibility in the command of the COA realization. However, this flexibility has two drawbacks for wargaming: First the evaluation of COA is dependent on human knowledge about the intricacies of the special simulation model and dependent on the human skills in the operation of the simulation. Second, interactive simulations are much slower than closed simulations (the difference can amount to several orders of magnitude). As a consequence of this limitation and of learning effects, it is seldom possible to make more than a few repetitions with the same (own and enemy) COA combination. Thus, there is little variation included into the evaluation. The parametric uncertainty of real combat is therefore often underestimated.

With closed simulation it is possible to make hundreds of simulation runs using different random numbers for all kind of elementary processes (movement, attrition, reconnaissance, communication etc.). Although this variation is somewhat compensated by its military "blindness" with respect to an interactive human commander, it can nevertheless capture a huge range of parametric uncertainty. The standard routine to take this uncertainty into consideration is the computation of measures of central tendency and dispersion (mean and variance, for example) (Ross 2002). For simplicity, let us assume that every simulation run ends with an exactly measurable result somewhere between a clear success (100) and a complete failure (0). A frequently used setting in standard military wargaming consists of three own COA which are compared with two enemy COA (usually the most likely and the most dangerous COA). A possible simplified result of a closed-simulation based statistical evaluation within this setting could look like Table 1.

Table 1: Simulation based Statistical Evaluation of COA (Example)

Most likely Enemy COA	Number of runs	mean	variance
COA 1	100	60	30
COA 2	100	55	20
COA 3	100	50	10

Most dangerous Enemy COA	Number of runs	mean	variance
COA 1	200	50	30
COA 2	200	40	10
COA 3	200	50	10

In order to get a definite result from these numbers it is necessary to weight between the two cases and the statistical measures. If we, for example, equally weight most likely and most dangerous enemy COA and ignore variance, COA 1 must be favoured because it has the highest average mean (55). A risk aversive decider would discount means by a certain proportion of the variance and would put a higher weight on the second case (the most dangerous enemy COA). This could lead to the preference of COA 3.

It is obvious that uncertainty in this approach is attributed as a statistical parameter to the given COA. The random effects in such a simulation system are produced by well understood random generators on the micro level of elementary process. However, the complicated interrelation and interaction of these processes can lead to a macro phenomenon (the overall combat, the outcome of a COA pair) which is astonishing. Since no combat simulation system can claim to be a valid representation of the future, the uncertainty measures created by this method are only measures of risk type 2 and should not be misinterpreted as measures of type1.

This standard reasoning can be easily transformed to a new method of dealing with modelling assumptions. Instead of analyzing the COA, explicit model assumptions are negated, implemented and the consequences of this negation analysed via stochastic simulation. The main methodological difference between the standard procedure and the special method of assumption-centred stochastic simulation is to take variance as decision criteria in the latter much more serious than in the former, since our focus of interest is the uncertainty attributed to the assumptions and not a disputable mean. The idea is to classify the analysed assumptions according to their criticality and plausibility (see Dewar 2002 and Hofmann 2007 for further information).

It should be mentioned that military simulation systems have reached an unprecedented level of complexity (Hofmann 2005). The example and reasoning presented are simplifications. Therefore, it now should have become clear, why we have chosen the aircraft example in section 4. It would have been extremely difficult to describe a wargaming example with the same completeness.

8. SIMULATION BASED WARGAMING FOR RISK TYPE 3 (STRUCTURAL UNCERTAINTY)

The basic idea of this approach is to use closed simulations *to detect* (not analyze) critical assumptions (and thereby actively dealing with structural uncertainty). The major advantage of closed simulations in comparison to interactive ones is, as already mentioned, the much greater speed of the former. It is therefore possible to run hundreds or even thousand of simulation runs within the time available for the decision making. The crucial question is, how *structural variability* can be introduced into the systems? By the use of random generators for the elementary process it is only possible to generate parametric variance. What is needed are random effects on the level of events and within the command and control modules. Random events can be easily generated if the demand for valid representations is completely given up. Then, a random event can be, for example, a regular event (detection, shot, etc.) without cause or a randomly chosen event from a historical data base. Introducing randomness into the command and control modules is somewhat more difficult, but nevertheless possible, if the notion of optimal behavior is neglected. By deliberately generating suboptimal behavior via random functions, the behavior of a command and control module becomes incalculable. Which is exactly what is intended. The key concept of the evaluation of such simulation runs is the exclusive debriefing of their extremes and the abdication of statistical reasoning. Disastrous simulation runs (from the perspective of the own planning) are taken as possible threats (implying critical assumptions), extremely advantageous runs are taken as possible chances. Means and measures of variance are not investigated, because the model is invalid anyway. All extreme simulation runs have to be checked by human experts, which can quickly discard them as completely implausible or further scrutinize them, because they appraise the chain of events as possible regardless its inconsequent creation within the simulation system. Their main function of this approach is to broaden the view of the planner with respect to unexpected future trajectories.

Invalid representations, deliberately modeling suboptimal behavior of automated forces and the renouncement of statistical evaluation for the benefit of mere extremes may seem absurd at first glance, but taken together and seen from the perspective of structural uncertainty they make perfect sense. Such simulations can be seen as explorations into the hidden realm of personal presumptions, that might be challenged by some extreme runs.

However, it is necessarily clear, that even this approach can not capture the whole range of possibilities spanned by real systems. Their major contribution in military (and maybe other kinds of) risk management might be, to open the decision maker's view to the completely unexpected by simply confronting him astonishing courses of action.

Reflecting this section it is now easily possible to explain the connection between the example in section 5 and simulation based tactical wargaming for structural uncertainty. Reliability functions that are non-monotonic in their components' reliability are extremely rare in practice and teaching. Thus, such an example will be astonishing for most engineers and students as well, recognizing their own presumptions about the subject (Hofmann and Lehmann 2007).

9. SUMMARY, CONCLUSION AND OUTLOOK

In most real applications risk is a multifaceted problem comprising objective and subjective probabilistic dangers as well as completely unknown threats. Some of the most critical uncertainties can be attributed to hidden personal presumptions and hitherto unquestioned modeling assumptions. Thus, an exhaustive methodology of risk management has to incorporate some strategies to cope with this challenge. In order to operationalize presumptions and assumptions we propose to see personal presumptions as part of what is called structural uncertainty, and the systematic questioning of explicit modeling assumptions as part of dealing with what is called parametric uncertainty.

With two simple examples we tried to demonstrate the importance of astonishing model/simulation results for the detection of (parametric model) assumptions and (structural) presumptions in the reliability of technical systems.

As a serious domain of application we subsequently introduced tactical wargaming. Two different methods of simulation and experimental design have been presented for parametric and structural uncertainty. The first approach follows traditional statistical reasoning for COA comparison, while stressing the importance of variance as opposed to means. In complete contrast to classical stochastic simulation and its evaluation by means of statistical reasoning, the second method is focused on the unexpected, uncommon, exceptional, ignoring validity and optimality. It is a method of thought triggering and definitely not of hard deduction.

Recent experiments during three major exercises have demonstrated the value of the unusual approaches, broadening the view on risk in general, and unexpected events in particular.

We are convinced that the methodologies do not only apply to military problems, but also to all kinds of other domains with parametric and structural uncertainty, including economic and industrial applications.

However, much work remains to do. First of all, more experiments have to be done in order to fine-tune the methods. Second, the value of the methods, especially the second one, heavily depend on unpredictable but sensible variability of the simulation system. For that purpose, much expertise has to be integrated into the respective modules. Third, we would really like to demonstrate the benefits of our approach in other domains.

REFERENCES

Barlow, R., Proschan, F., 1975. *Statistical Theory and Life Testing (Probability Models)*, New York: Holt, Rinehart and Winston.

Clausewitz, C. v., 1832. *Vom Kriege*. Reinbek: Rowohlt Tb. 1978.(also available on www.clausewitz.com)

Dewar, J. A., 2002 *Assumption-Based Planning: A Tool for Reducing Avoidable surprises*. Cambridge UK: Cambridge Press.

Gigerenzer, G., Selten, R., 2002. *Bounded Rationality: The Adaptive Toolbox*. Massachusetts : MIT Press.

Hofmann, M. 2003. Modeling Assumptions: How they affect Validation and Interoperability. *Proceedings of the 5. European Simulation Interoperability Workshop*, Stockholm.

Hofmann, M., 2005. On the Complexity of Parameter Calibration in Simulation Models. In: *Journal of Defense Modeling and Simulation 2, (4).*

Hofmann, M., Lehmann, T. 2007. On simulation-based Wargaming: Comparison of two different methodological approaches. *Information and Security.* Vol. 22, 64-72.

Hofmann, M., Lehmann, T., 2007. *Tactical Wargaming, the Coup D' Oeil and IT-Support-Systems*. Third International Conference On Military Technology Miltech 3, 129-146, Stockholm, Sweden.

Hofmann, M., Junge, B., 2008. *Dealing with Structural Uncertainty in Tactical Wargaming*. In: *Proceedings of the 9. European Simulation Interoperability Workshop*, Edinburgh, Great Britain.

Knight, F. H., 1921. *Risk, uncertainty and profit.* Hart, Schaffner & Marx, MA: Houghton Mifflin Company Boston.

Rohatgi, V. K., 1976. *An Introduction to Probability Theory and Mathematical Statistics.* New York: John Wiley & Sons.

Ross., S. M., 2002. *Simulation*. 3rd ed., San Diego: Academic Press.

Stachowiak, H., 1973. *Allgemeine Modelltheorie*., Wien, New York: Springer Verlag.

AUTHORS BIOGRAPHIES

MARKO HOFMANN is Project Manager at the Institut für Technik Intelligenter Systeme (ITIS e.V.) an der Universität der Bundeswehr München, Germany. After his studies of computer science at the Universität der Bundeswehr München, he served two years in an army battalion staff. From 1995 to 2000 he was research assistant at the Institute for Applied System Analysis and Operations Research (IASFOR), where he got his Ph. D. in computer science. Since April 2000 he is responsible for basic research in applied computer science at the Institut für Technik Intelligenter Systeme (ITIS e.V.). He gives lectures in operations research, computer science and mathematics at the Universität der Bundeswehr München and at the University of Applied Science in Kufstein.

THOMAS KRIEGER is a Research Assistant at the Institut für Technik Intelligenter Systeme (ITIS e.V.) an der Universität der Bundeswehr München, Germany. He received his Diploma in Mathematics in 2000 from the Technische Universität Dresden, Germany and his Doctorate in Mathematics in 2003 from the Universität der Bundeswehr München. His research interests include Modelling and Simulation as well as Game Theory.

MODELLING AND SIMULATION OF AUTONOMOUS LOGISTIC PROCESSES

Bernd Scholz-Reiter[(a)], Torsten Hildebrandt[(a)], Jan Kolditz [(a)]

[(a)] Planning and Control of Production Systems, University of Bremen, Germany

[(a)] {bsr, hil, kol}@biba.uni-bremen.de

ABSTRACT

Autonomous control of logistic processes is proposed as a means for enterprises to better face dynamics and complexity, caused e.g. by globalization. This conference contribution will first briefly sketch the idea of autonomous control of logistic processes. Second it will discuss existing modelling approaches on the basis of requirements to a suitable method and subsequently will outline our modelling method designed for engineering of autonomous logistic processes. The third part will detail the transformation of conceptual models constructed by using this method into a simulation on the basis of an industrial case study. Moreover some results of simulation studies will be presented and discussed. The paper is concluded by a summary and an outlook on future work.

Keywords: autonomous control, process modelling, agent oriented simulation, modelling method

1. INTRODUCTION

Coping with complexity of logistic systems is a task undertaken only insufficiently today. One reason is that the centralised planning and control strategies used presume information that is mostly not available in the required quality and quantity. Autonomous control within the context of the German interdisciplinary research effort CRC 637 means processes of decentralized decision making of interacting system elements in heterarchical structures. Concretised towards autonomous control of logistic processes it is defined as "[…] characterized by the ability of logistic objects to process information, to render and to execute decisions on their own" (Huelsmann and Windt 2007). A logistic object fulfilling this definition is called an autonomous logistic object; to support its design implicates an approach focused on these objects. RFID (radio frequency identification)- and smart label-technologies and their successors in the foreseeable future are seen as an enabling technology to realize autonomous control of logistic processes.

The design of autonomous logistic processes is an extensive task, whose complexity normally impedes its complete notional analysis and design. Therefore it is necessary to utilise construction of models where in particular graphical methods allow descriptive and easily comprehensible models. For supporting such a modelling task on one hand methods from business process modelling and on the other hand methods for software agent modelling should be considered.

In section 2 this paper sketches the engineering of a system based on autonomous logistic processes and discuss different existing modelling approaches on the basis of our specific requirements in that context. In section 3 we outline a modelling method specifically designed for conceptual modelling during the engineering phase of systems based on autonomous logistic processes.

Section 4 presents an industrial case study transforming the conceptual model of a manufacturer of automotive supplies into a simulation model. The subsequent sections 5 and 6 present results from a simulation study.

2. MODELLING IN THE CONTEXT OF ENGINEERING SYSTEMS BASED ON AUTONOMOUS LOGISTIC PROCESSES

2.1. Engineering autonomous logistic processes and requirements on an adequate modelling method

On the basis of the general Systems Engineering procedure model (Haberfellner 2002) the engineering of an autonomous logistic system can be described by the phases initiation, preliminary study, specification, simulation, infrastructure configuration, cost benefit estimation, establishment and introduction (Scholz-Reiter et al. 2007). The four phases from specification to cost benefit estimation form the methodical core as an iterative process.

In the specification phase a conceptual model of the system is created in the form of a semi-formal specification of the autonomous logistic objects. Moreover identification, design and allocation of decision processes are performed. It has to be clarified which elements are part of the system and which of them are "intelligent" respectively autonomous entities.

During the simulation phase the design created before is tested. Especially operability and impact on logistics performance of the whole system are focused here. This step therefore allows an effective comparison to the existing or alternative ways of controlling the examined logistic system. The simulation code may already be part of the engineering process of the planned control software if the code is reusable. Otherwise the core software engineering process starts in the implementation phase.

On the basis of the insights gained before an estimation of needed hardware equipment for the autonomous system (for example what kind of communication infrastructure) can be made, getting more detailed with every iteration loop. For example from allocation of control processes and data packets to entities of the logistic system necessary memory and computing capacity may be derived.

Every iteration is concluded by a cost benefit estimation. On the basis of the rating and subsequent decision the original process model can be adjusted according to the new conclusions. In case of repeated negative results in this step an application of autonomous control has to be abandoned for this scenario.

Using the definition of autonomous logistic processes and the necessary phases for engineering a system based on this principle four main requirements can be formulated on a methodology for modelling autonomous logistic processes.

- The methodology has to fulfil the general definitional attributes of a modelling methodology. Therefore it must imply at least a notation, a procedure model and a fundamental structuring like a view concept (definition orientation).
- The methodology must be application area oriented and therefore has to aim first at planning and control of logistic systems and second at modelling the constitutive characteristics of autonomy in logistics (application area orientation).
- The methodology must be appropriate for the user, for which reason an explicit consideration of logistic domain experts has to be assured (user orientation).
- The methodology must consider the use of models created with it, what especially requires consideration of subsequent agent oriented software implementation (model usage orientation).

2.2. Evaluation of existing approaches

In this subsection existing approaches for agent oriented modelling as well as process and logistic oriented modelling are examined concerning their appropriateness for modelling autonomous logistic processes. For the evaluation especially the four main requirements formulated before are relevant. In table 1 the evaluation is summarised by comparing methods and requirements. In case of a positive rating (+) the requirement is predominantly fulfilled, in case of a neutral rating (0) the requirement is partly fulfilled and in case of a negative rating (-) the requirement is inadequately fulfilled.

Regarding the software implementation the concept of agent-oriented software engineering is very close to the paradigm of autonomy in logistics due to the attributes of a software agent (Wooldridge and Jennings 1995) like autonomy, reactivity or adaptivity. Important agent oriented methods are Gaia (Cernuzzi et al. 2004), MaSE (DeLoach et al. 2004), MAS-CommonKADS (Iglesias et al. 1998), Tropos (Bresciani et al. 2004) and DACS (Bussmann et al. 2004). However in spite of the numerous existing methodologies for agent oriented software engineering, the deficits in connecting the software engineering with real production systems or with industrial systems in general is seen as one cause for the relatively low number of agent based systems actually used in industry (Monostori et al. 2006), (Hall et al. 2005). For Holonic Manufacturing Systems (HMS) (Valckenaers et al. 1999), which can be seen as an important approach to autonomy (Windt 2006), a significant demand for methods based on software engineering principles is seen, which support the designer of the HMS software system in all stages of the development process (Giret and Botti 2005), (McFarlance and Bussmann 2003). A main aspect of the insufficient methodical support is the requirements analysis and thus the linkage between real scenario and HMS-based software system (Giret and Botti 2006). In general agent-oriented software engineering methodologies accentuate important aspects like autonomy but widely disregard the decisions (Bussmann et al. 2004) being a constitutive characteristic of autonomous logistics processes. Moreover according to their intended use they focus on a detailed design of a software system but disregard the integration of a logistics domain expert in the specification of the system.

Table 1: Comparing Requirements and Existing Modelling Approaches

	ARIS	CIMOSA	IUM	SOM	MPSF	Gaia	MaSE	MAS-CommonKADS	Tropos	DACS
definition orientation	+	+	+	+	0	+	+	+	+	0
application area orientation	-	-	-	-	0	0	0	0	0	0
user orientation	+	0	+	+	+	-	-	-	-	+
model usage orientation	0	0	0	0	0	+	+	+	+	+

The lacking consideration of domain experts does not apply to the Methodology for Designing Agent Based Production Control Systems (DACS) (Bussmann et al. 2004). DACS is meant to support a production engineer without experience in software agent technology during design of an agent based production control system.

In DACS no standardised notation is used but so-called trigger diagrams. These easily get very complex and unclear and are no longer presentable in a single diagram, but there are no possibilities mentioned to decompose. The central step of the DACS procedure is the agent identification. In most cases this step should lead to an aggregation of decisions in agents that represent physical components of the production system. That assignment of agents to physical elements is also a result of the case study and is endorsed by the used principle to identify things rather than function (Parunak et al. 1998). But this puts the extensive process of decomposing and composing of decision tasks into question. Rather using the physical components as orientation from the beginning seems to be more adequate. Furthermore it is not evident that the final definition of the agents to be used should be made by a production engineer who is not experienced in agent technology at all. For example it may be necessary to split up tasks because of limited functions of a single agent and therefore to partly abandon the physical component orientation. Nevertheless this orientation is reasonable especially during conceptual design even if more agents are responsible for the tasks of a single physical component. Moreover DACS does not support an iterative process, what would be necessary for the design of a system with realistic complexity.

Important process modelling methods are ARIS (Scheer 2000), CIMOSA (Vernadat 2006), IUM (Mertins and Jaekel 2006), SOM (Ferstl and Sinz 2006) and MPSF (Dangelmaier 2001). In context of software engineering, methodologies for business process modelling are intended to support the development of centralised information systems (Scheer 2000). Because of this purpose dedicated concepts for specifying decentralised approaches are missing. In principal these can be included by adjusting existing reference models (Boese and Windt 2007), but still there is a lack of sufficient instruments for explicit modelling of communication processes and protocols. However the most important aspect is that there is no guidance for designing a logistic system under consideration of autonomic control strategies and therefore no dedicated procedure model exists. On one hand methodologies for process modelling like ARIS or IUM do not take special care about design of planning and control processes. On the other hand the concept for model-based planning and control of production systems (MPSF) aims at the design of planning and control processes, but there a strict hierarchical and centralised approach is pursued.

According to these aspects the modelling methodology in context of engineering autonomous logistic systems shall be the connection between real-world oriented business process modelling and agent-oriented software engineering for the specific domain. The specification should focus on the planning and control processes of the real system or the system to be realised respectively. However the constructed model shall to some extend still be independent from the detailed software design. For example the logistic objects in an autonomous logistic system like machines, commodities or conveyors may be modelled as single autonomous entities, but the software architecture may differ. This flexibility allows the software engineer to split up abilities of a logistic object on multiple software agents when this is required because of favoured agent architectures or practical limits of a single agent. These activities of specifying the control processes on one hand and of designing the software system on the other hand require different qualifications. Thus a software engineer is in charge of the software design and therefore the determination of the software agent architecture. In contrast a logistics domain expert specifying the autonomous logistic system is responsible for planning and control processes and constructs a model that formulates requirements to the software system. When several people with different qualifications are involved in engineering a system, a modelling notation that is persistently used from the process model of the system to the implementation of the software avoids a gap in the engineering process by using standardised semantic concepts in the different disciplines (Specker 2005). One possibility for this is the use of the Unified Modeling Language (UML). As a graphical, semi-formal notation it is broadly used - besides software development (especially agent-oriented approaches are of particular interest here, see for instance AUML (Odell et al. 2001, Bauer et al. 2001) it is also used for knowledge modelling (Schreiber et al. 2001) or business process modelling (Oestereich et al. 2003).

3. METHOD FOR CONCEPTUAL MODELLING OF AUTONOMOUS LOGISTIC PROCESSES

In this section the first phase of the modelling cycle, the semi-formal specification of the autonomous logistic system, is focused. To support this specification a modelling methodology as part of the Autonomous Logistics Engineering Methodology (ALEM), with its components ALEM-N (ALEM-Notation), ALEM-P (ALEM-Procedure) and ALEM-T (ALEM-Tool), is proposed. ALEM-N consists of a view concept comprised of views each showing specific aspects of the logistic system as well as the notational elements to be used in each view and their intended meaning. ALEM-P is a procedure model describing the steps to be followed in generating a model and is intended to guide the user through analysis and specification of an autonomously controlled logistic system. ALEM-T is a software tool, specifically tailored to support the notation and the procedure model.

3.1. View Concept and Notation

Creating process models usually leads to a high degree of complexity. A view concept serves as a means to reduce the complexity constructing a model (Scheer 2000). ALEM-N includes a view concept for modelling of autonomous logistic processes, distinguishing five different views. Moreover a notation primarily based on a selection of UML notational elements is part of ALEM-N. In the view concept a fundamental distinction is made between a static and dynamic (sub-) model. The static model describes the structure, the dynamic model the behaviour of the modelled system, following the basic distinction in UML (Unified Modelling Language, OMG (2006)) that is also appropriate here.

The Structure View showing the relevant logistic objects is the starting point. The basic elements for this view are UML class diagrams. Besides objects and classes the structure view can show relationships between them, for instance in the form of associations or inheritance relationships.

The Knowledge View describes the knowledge, which has to be present in the logistic objects to allow a decentralized decision making. This view focuses on composition and static distribution of the knowledge while not addressing temporal aspects. For this purpose UML class diagrams and Knowledge Maps are sufficient.

The Ability View depicts the abilities of the individual logistic objects. Processes of a logistic system need certain abilities, which have to be provided by the logistic objects. These abilities are supposed to be seen as abstractions of problem types and their solving capabilities occurring in reality.

The Process View depicts the logic-temporal sequence of activities and states of the logistic objects. Here the objects' decision processes can be modelled. The notation elements used for this are activity diagrams as well as state diagrams.

The Communication View presents the contents and temporal sequence of information exchange between logistic objects. To display the communication UML sequence diagrams showing the interacting partners, the messages and their temporal progression as well as class diagrams to display communication contents are supposed to be used.

The connections between the views are based on the ones defined in the UML meta model (OMG 2006) but are also extended to better guarantee model consistency while additionally restricting freedom during model construction.

3.2. Procedure Model

The procedure model ALEM-P is a guideline for modelling autonomous logistic processes, which contributes on one hand to the assurance of model quality and on the other hand to the reduction of the effort during model construction. It is a specific procedure model, which recommends operational activities using the notational elements and concepts described before. Thereby a system modeller with deepened knowledge about logistics planning and control is enabled to construct a semi-formal system specification to support analysis, design and improvement of systems based on autonomous control.

The procedure model defines steps to pass during model construction, therein activities to perform and results to get out of every step. Furthermore methods and instruments are recommended to support the work. Among these are firstly the presented view concept and diagrams, secondly modelling conventions in terms of construction and consistency rules and thirdly existing techniques suitable for the individual steps. Additionally there are indicators given for necessary iterations that may be initiated in a step, which cause a reengineering cycle by referring to a former step. Basically the procedure is inspired by the top down principle because the system and the enclosed processes are examined on a rather abstract level before they are detailed and concretised. However the focus on selected autonomous logistic objects and their reciprocal coordination with each other as well as the other system elements involves a high importance of the bottom up principle. Thus the procedure is a combination of top down and bottom up approach.

The first step in the specification procedure for autonomous logistic systems broaches the issue of objectives in the system. Starting point are the global system objectives. For a production system the classic goals of production logistics shall be used, from which more concrete local goals can be derived. The documentation of objectives is done in the knowledge view by using class diagrams.

The second step of the specification procedure is the design of the system structure and therewith the collection and documentation of the system elements and their static relations. Central to this step are the autonomous logistic objects - the modeller has to plan which system elements shall have autonomous abilities and which ones not. This aspect will afterwards be further elaborated in the next step. The modelling of the structure is done in the structure view using class diagrams.

The third step of the modelling procedure aims at a structuring of abilities and their mapping to the different logistic objects. Abilities are interpreted as abstract collections of operations that enable an autonomous logistic object to perform certain planning and control tasks. Abilities are modelled in the ability view using class diagrams and especially the concept of interfaces.

The fourth step concentrates on the modelling of the processes running in the system, especially the necessary control processes. The process design is separated in two sub-steps. First routine processes assuming a progression without disturbances are modelled and afterwards these are systematically complemented by processes for handling disturbances and unplanned events. The modelling is done in the process view using activity diagrams and state machines.

The fifth step of the modelling procedure focuses on the decisions. To support identification and adequate description of decisions the structuring of a decision model from decision theory is adapted here (Bussmann 2004, Laux 2005). Thus a control decision can be characterised by a decision maker, an objective and a decision rule representing the objective, a trigger as well as a decision space. The modelling is done in the process view, in particular using activity diagrams.

In the sixth step the focus lies on the knowledge needed for decision making. For that purpose every decision has to be analysed what knowledge is needed. The explicit consideration in the process model is carried out in activity diagrams using object nodes. After examining what knowledge is needed, it has to be specified where it comes from by allocating the knowledge objects. The important point is not the location of knowledge usage, what has been relevant during examination of the decision processes. In contrast it has to be specified where the knowledge objects are available in constantly updated form and thus where demanding autonomous logistic objects can access it. For modelling the allocation of information objects knowledge maps are used.

The communication is modelled in step seven. Thereby two main aspects have to be distinguished. On one hand there are the communication processes and on the other hand the exchanged messages. The modelling of communication is done using sequence and class diagrams. An example for a sequence diagram is shown in figure 1. This communication protocol between machines and commodities is used for the allocation of commodities to machines. The figure shows the sequence of message exchange where the commodity requests a machining operation answered by the machine with a quote containing the possible completion date. The commodity selects one machine and sends an notification after arrival. After the machine has checked its own ability of processing the arrived commodity it sends an arrival notification or a refusal of acceptance.

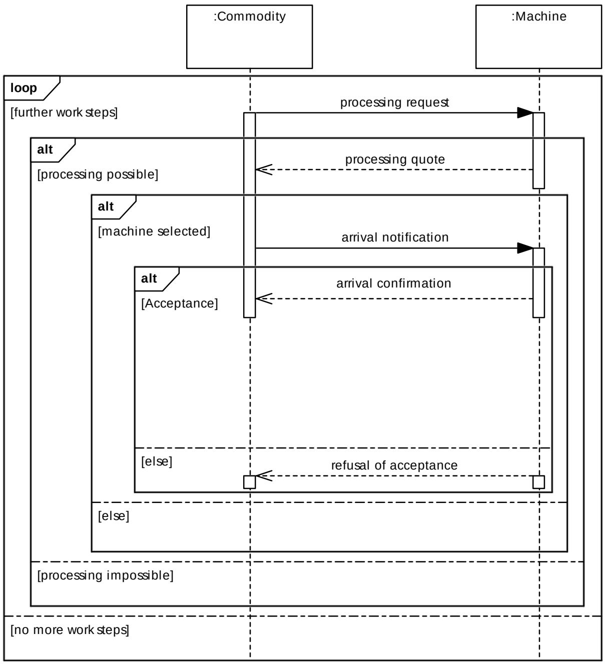


Figure 1: Sequence Diagram for the Allocation of Commodities to Machines

In the eighth step the concrete scenario data is collected. For the classes defined during the previous steps all instances have to be documented to form the basis for the succeeding simulation phase and in the end for the operability of the system. The data is entered in simple lists or matrices. Moreover in ALEM-T it is possible to show at least the resources of the logistic system in a layout diagram and to enter the data there.

4. CASE STUDY

We applied the modelling method just presented to an industrial case study from a manufacturer of automobile supplies. Following the eight step-procedure to derive a model we developed a to-be concept for the control of this production logistic system based on the principle of autonomous control.

In order to assess the operability of the model and it’s logistic performance we derived a multi-agent-based simulation model to simulate the scenario, both without and also with the influence of disturbances. To implement the simulation model we used the multi-agent simulation environment SeSAm (Shell for Simulated Agent Systems) (Kluegl 2006). The most important reason for choosing SeSAm was that it provides visual agent modelling using an UML-based notation. Thus further usage of conceptual models constructed during the specification phase is eased and ensured. Unfortunately the performance of the created simulation model turned out not to be sufficient to run the experiments we intended. Especially determining proper buffer/stock sizes (see the text below) required numerous simulation runs. To be able to perform these simulations we implemented the model and production control methods in our own, Java-based discrete event simulation kernel.

The simulation model consists of 34 machines, 10 different products are manufactured. The material flows within the system are re-entrant, i.e. products have to visit certain machines more than once. Furthermore on some machines sequence-dependent setup times occur. There are further restrictions on machine capabilities: there are alternative machines for most production steps, but there are also restrictions regarding the product variants each machine can produce.

The company our case study is based on produces valve spring holders. A valve spring holder couples the valve spring with the valve in an automobile engine and serves as support for the valve spring. The production of valve springs includes several production steps with different production technologies (see figure 2). For our simulations we selected 10 out of 100 different variants of valve spring holders on the basis of an ABC analysis that cover around 75% of the overall production volume. The differences of the product variants in size and/or material result in different processing times on some production steps, especially in off-pressing, annealing and pressing. So the overall processing time varies between 945 minutes and 1045 minutes per standard lot. Moreover setup times have to be taken in account on off-pressing and pressing machines as well as on packing machines when different product variants are processed successively.

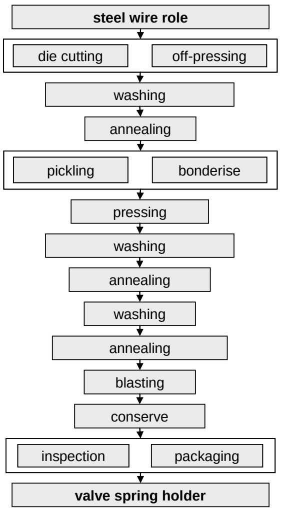


Figure 2: Case Study Production Process

The allocation of commodities to the machines is done as follows. Before each production step a commodity requests processing quotes from all potential machines for the next work step. Machines check the requests and in case of suitability calculate possible processing dates depending on their state and schedule and send back processing quotes containing an estimated time when the requesting commodity can be finished. A commodity selects the machine with the earliest possible date. In the following we refer to this method of production control as *Method A.*

We compare this method with a variant of the well-known KANBAN method (Ohno 1988), in the following referred to as *Method B.* KANBAN was chosen because it is also a decentralized and widely known method for production control. It is furthermore quite simple to implement – its use is even possible without IT-support.

We aim at a zero lead time for customer orders, i.e. arriving customer orders should be processed with a very low flow time close to zero. To minimise costs these low flow times have to be achieved with a work in progress (WIP)-level as low as possible. To achieve this, we have to establish a stock of proper size for finished goods for Method A and properly dimensioned buffers for Method B. The sizes finally used for the results in section 6 are given in table 2, “Stock Size” is the stock size of finished goods used for Method A, the KANBAN buffer sizes for Method B are shown in column 4. The column for KANBAN only contains the sum of the buffer sizes over all 12 production steps of the respective product. Stocks and buffers were dimensioned to achieve a very low mean flow time at a long-term expected utilization of 85% (no downtimes).

To asses the performance of our modelled and implemented method A, we considered two factors, reflecting external or internal sources of disturbances: different demand levels and machine downtimes. Different demand levels are simulated by different bottleneck utilizations of 80, 85, 90 and 95%. Machine breakdowns are considered with three levels: no downtimes, mean time between failures 2 days, mean time between failures 1 day. For the latter two settings we use an exponential distribution with the respective mean.

5. SIMULATION METHODOLOGY

As already stated we simulate 34 machines organized in 10 machine groups. 2 machine groups have to be visited twice so there are 12 operations to finish each lot. We assume a fixed product mix (table 2) but actual demand can vary between 80 and 95% bottleneck utilization based on the product mix shown in table 2. Inter-arrival times are exponentially distributed.

Table 2: Product Mix and Buffer/Stock Size used

Product	Frequency	Stock Size	Kanban Buffer Sizes (Sum)
1	18,1%	14	18
2	16,8%	14	18
3	13,1%	11	13
4	12,3%	10	11
5	11,2%	11	13
6	8,7%	10	11
7	5,4%	8	8
8	5,2%	7	8
9	4,9%	7	7
10	4,3%	6	7
Sum	100,0%	98	114

We are interested in steady state performance of the production system and our main concern is the mean flow time of a lot, i.e. the time required from the arrival of a customer order until the time the order can be fulfilled, as well as the WIP-level in the factory, measured in number of lots.

We therefore simulate a time span of 5 years of continuous production, but to avoid bias only the last 4 years are used to produce the results. We perform at least 5 independent replications of our simulation experiments and if required further replications until we get a confidence interval of at most ± 1% mean flow time at a confidence level of 95% (Law 2007) or alternatively ± 1 minute, whichever is larger.

6. SIMULATION RESULTS

Using the experimental setup just described we achieved results concerning WIP levels and flow times as shown in figure 3 and table 3 respectively.

As can be seen in figure 4 WIP levels for Method A are constant and independent of utilization or breakdowns. It is always equal to the sum of the stock sizes (see table 2), because as soon as a finished good is removed from stock, a new lot is started to refill the stock – the total number of lots on the shop floor and in the stock remains the same. For method B, KANBAN, however WIP decreases with an increasing utilization and increasing frequency of machine breakdowns due to an increasing number of buffer spaces awaiting to be refilled.

Concerning flow times we achieved the results of table 3. For the factor combination used to determine buffer/stock sizes (85% utilization, no breakdowns) both method A and method B show very low results (nevertheless method A needs a lower WIP-level to achieve this. Method A, based on autonomous control, however is clearly more robust with respect to an increasing utilization and more frequently occurring machine breakdowns.

Another criterion is the effort required to find proper parameters for the production control methods. Here again method A outperforms method B as it only requires 10 parameters (the stock size for each finished good) to be set, whereas for KANBAN the size of 120 KANBAN buffers (10x12, one potential buffer for each product and production step) has to be set appropriately.

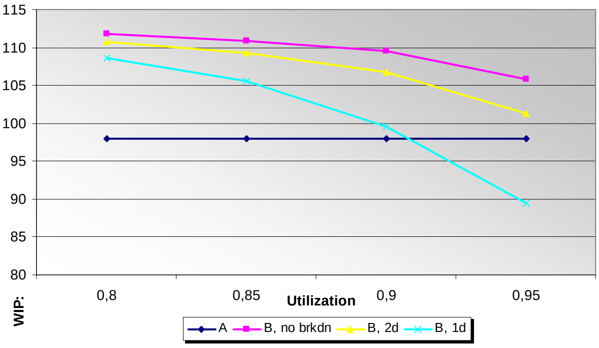


Figure 3: WIP Levels Achieved

Table 3: Flow Times (in Minutes) for Different Settings of Breakdowns and Utilizations

		Method A			Method B		
		none	**2d**	**1d**	**none**	**2d**	**1d**
Utilization	**80%**	0	0	2	1	3	12
	85%	1	1	10	3	8	58
	90%	2	7	85	9	37	393
	95%	33	87	1709	95	246	8432

7. SUMMARY AND OUTLOOK

This paper addressed the topic of modelling autonomous logistic processes. Therefore, after a short definition of autonomous control in the context of logistics, we discussed requirements to a modelling method in this context and evaluated existing methods for this purpose. Afterwards we briefly sketched our modelling method ALEM, specially targeting the design of autonomous logistic processes.

We applied our method to an industrial scenario (section 4) and used this model to derive a simulation model to asses operability and logistic performance of the modelled system. In section 6 we compared the system's performance with the well-known KANBAN-method for production control.

Further work will concentrate on further integrating the modelling and simulation phases and allow a semi-automatic transformation of the conceptual ALEM-N-model into a simulation model, a step currently performed manually. Furthermore we plan to further improve our modelling method by applying it to additional industrial scenarios.

ACKNOWLEDGMENTS

This research is funded by the German Research Foundation (DFG) as part of the Collaborative Research Centre 637 Autonomous Cooperating Logistic Processes - A Paradigm Shift and its Limitations (SFB 637).

REFERENCES

Bauer, B., Mueller, J.P., Odell, J., 2001. Agent UML: A formalism for specifying multiagent software systems. *International Journal of Software Engineering and Knowledge Engineering*. 11 (3): 207–230.

Boese, F., Windt, K., 2007. Business Process Modelling of Autonomously Controlled Production Systems. In: M. Huelsmann, K. Windt, eds. *Understanding Autonomous Cooperation & Control in Logistics - The Impact on Management, Information, Communication and Material Flow*. Berlin: Springer, 73–83.

Bresciani, P, Giorgini, P., Giunchiglia, F., Mylopoulos, J., Perini, A., 2004. Tropos: An Agent-Oriented Software Development Methodology. *International Journal of Autonomous Agents and Multi Agent Systems*, 8 (3): 203–236.

Bussmann, S., Jennings, N.R., Wooldridge, M., 2004. *Multiagent Systems for Manufacturing Control. A Design Methodology*. Berlin: Springer.

Cernuzzi, L, Juan, T., Sterling, L., Zambonelli, F., 2004. The Gaia Methodology. In: F. Bergenti, ed. *Methodologies and Software Engineering for Agent Systems. The Agent-Oriented Software Engineering Handbook*. Boston: Kluwer Academic Publishers, 69–88.

Dangelmaier, W., 2001. *Fertigungsplanung - Planung von Aufbau und Ablauf der Fertigung. Grundlagen, Algorithmen und Beispiele*. Berlin: Springer.

DeLoach, S.A., 2004. The MaSE Methodology. In: F. Bergenti, ed. *Methodologies and Software Engineering for Agent Systems. The Agent-Oriented Software Engineering Handbook*. Boston: Kluwer Academic Publishers, 107–126.

Ferstl, O.K., Sinz, E.J., 2006. *Grundlagen der Wirtschaftsinformatik*. Muenchen: Oldenbourg.

Giret, A., Botti, V., Valero, S., 2005. MAS Methodology for HMS. In: V. Marik, ed. *Holonic and Multi-Agent Systems for Manufacturing. Proceedings of 2nd International Conference on Industrial Applications of Holonic and Multi-Agent Systems, HoloMAS 2005*, 39–49. August 22-24 2005, Copenhagen, Denmark.

Giret, A., Botti, V., 2006. From system requirements to holonic manufacturing system analysis. *International Journal of Production Research*. 44(18-19): 3917–3928.

Haberfellner, R., 2002. *Systems Engineering*. Zuerich, Switzerland: Industrielle Organisation.

Hall, K.H., Staron, R.J., Vrba, P., 2005. Experience with Holonic and Agent-Based Control Systems and Their Adoption by Industry. In: V. Marik, ed. *Holonic and Multi-Agent Systems for Manufacturing. Proceedings of Second International Conference on Industrial Applications of Holonic and Multi-Agent Systems, HoloMAS 2005*, 1–10. August 22-24 2005, Copenhagen, Denmark.

Huelsmann, M., Windt, K., 2007. *Understanding Autonomous Cooperation & Control in Logistics – The Impact on Management, Information, Communication and Material Flow*. Berlin, Germany: Springer.

Iglesias, C.A., Garijo, M., Gonzalez, J.C., Velasco, J.R., 1998. Analysis and Design of Multiagent Systems Using MASCommonKADS. In: M.P. Singh, ed. *Proceedings of the 4th International Workshop on Agent Theories, Architectures, and Languages (ATAL'97)*, 313–327. July 24-26, 1997 Rhode Island, USA.

Kluegl, F., Herrler, R., Fehler, M., 2006. SeSAm: Implementation of Agent-Based Simulation Using Visual Programming. In: *Proceedings of the AAMAS 2006*, 1439-1440. May 8-12 2006 Hakodate, Japan.

Laux, H., 2005. *Entscheidungstheorie*. Berlin: Springer.

Law, A., 2007. *Simulation Modeling and Analysis*, 4th edition. Boston et al.: McGraw-Hill.

McFarlance, D.C., Bussmann, S., 2003. Holonic manufacturing control: rationales, developments and open issues. In: S.M. Deen, ed. *Agent-Based Manufacturing - Advances in the Holonic Approach*, Berlin, Germany: Springer, 303–326.

Mertins, K, Jaekel, F.-W., 2006. MO2GO: User Oriented Enterprise Models for Organisational and IT Solutions. In: P. Bernus, ed., *Handbook on Architectures of Information Systems*. Berlin, Germany: Springer, 649–663.

Monostori, L., Vancza, K., Kumara S.R.T., 2006. Agent-Based Systems for Manufacturing. *CIRP Annals*, 55(2): 697–720.

Odell, J.J., Parunak, H.V.D., Bauer, B., 2001. Representing agent interaction protocols in UML. In: P. Ciancarini, M. Wooldridge, eds. *Agent Oriented Software Engineering*. Berlin: Springer, 121-140.

Oestereich, B., Weiss, C., Schröder, C., Weilkiens, T., Lenhard, A., 2003. *Objektorientierte Geschäftsprozessmodellierung mit der UML*. Heidelberg: Dpunkt.

OMG, 2006. *Unified Modeling Language Specification, Version 2.0*. Available from: www.uml.org (accessed 01 December 2006).

Parunak, H. Van D., Sauter, J., Clark, S., 1998. Toward the Specification and Design of Industrial Synthetic Ecosystems. In: M.P. Singh, ed. *Intelligent Agents IV: Agent Theories, Architectures, and Languages*. Berlin: Springer, 45–59.

Scheer, A.-W., 2000. *ARIS - Business Process Modeling*. Berlin: Springer.

Scholz-Reiter, B., Kolditz, J., Hildebrandt, T., 2007. Engineering autonomously controlled logistic systems. *International Journal of Production Research:* 1-20.

Schreiber, G., Akkermans, H., Anjwerden, A., de Hoog, R., Shadbolt, N., van de Velde, W., Wielinga, B., 2001. *Knowledge Engineering and Management - The CommonKADS Methodology*. Cambridge, USA: MIT Press.

Specker, A., 2005. *Modellierung von Informationssystemen. Ein methodischer Leitfaden zur Projektabwicklung.* Zuerich: vdf Hochschulverlag.

Ohno, T., 1988. *Toyota production system: beyond large-scale production,* Portland, USA: Productivity Press.

Valckenaers, P., Heikkilä, T., Baumgaertel, H., McFarlane, D., Courtois, J.-P., 1999. Towards a novel manufacturing control principle. In: H. Van Brussel, ed. *Proceedings of the Second International Workshop on Intelligent Manufacturing Systems*, 871–875. September 22-24, 1999 Leuven, Belgium.

Vernadat, F., 2006. The CIMOSA Languages. In: P. Bernus, ed. *Handbook on Architectures of Information Systems*. Berlin: Springer, 251–272.

Windt, K., 2006. Selbststeuerung intelligenter Objekte in der Logistik. In: M. Vec, ed. *Selbstorganisation - Ein Denksystem fuer Natur und Gesellschaft,* Cologne, Germany: Boehlau, 271–314.

Wooldridge, M. and Jennings, N.R., 1995. Intelligent agents: theory and practice. *The Knowledge Engineering Review* 2 (10): 115–152.

AN ANALYTICAL APPROACH FOR OPTIMISING THE NUMBER OF REPAIRMEN FOR LARGE SCALE, HOMOGENEOUS, MULTI-SERVER SYSTEMS

Orhan Gemikonakli [(a)], Hadi Sanei [(b)], Enver Ever [(c)], Altan Koçyiğit [(d)]

[(a)] School of Computing Science, Middlesex University, The Burroughs, London NW4 4BT, UK
[(b)] Halcrow Group Limited, TBG, Vineyard House, 44 Brook Green, London W6 7BY, UK
[(c)] School of Engineering, Design and Technology, University of Bradford, Richmond Road, Bradford, BD7 1DP, UK
[(d)] Informatics Institute, Middle East Technical University, 06531, ODTU, Ankara, Turkey

[(a)] o.gemikonakli@mdx.ac.uk, [(b)] saneih@halcrow.com, [(c)] eever@bradford.ac.uk, [(d)] kocyigit@metu.edu.tr

ABSTRACT

Fault tolerant, large scale multi-server systems require an optimum number of repairmen for maximising performability. However, performability evaluation of such systems is difficult due to the state space explosion problem. In this paper, a simple and flexible approximate technique capable of overcoming state space explosion problem in computing the performability of large Markov models is presented. For validation of results, simulation has been used. It is shown that, this approach can handle large state space. The proposed method allows analysing the breakdown repair behaviour of large multi server systems. An optimisation study is presented together with numerical results showing the relationship of the number of servers and the number of repairmen for optimum performance.

Keywords: Large scale multi-server systems, State Explosion problem, Markov Process, Performability and Simulation.

1. INTRODUCTION

The number of repairmen in a fault tolerant multi-server system can affect the system's overall performability significantly. When multi-server systems are considered, some of the servers in the system may be redundant because of poor repair facilities. The availability of the servers highly depends on the ratio of failure and repair rates. The probability of failures linearly increases with the number of servers employed. It is necessary to address the effects of server availability on overall performability of large scale systems. In this paper large scale, homogeneous multi-server systems are considered. The queuing model under study is a homogeneous multi server system with unreliable servers and a common queue. Homogeneous multi server systems with unreliable servers have been considered in (Chakka and Mitrani 1992; Boxma et al. 1994; Chakka et al. 2002; Mitrani 2005). However a solution method to obtain the steady state probabilities of large scale homogeneous multi server systems which is applicable to both loaded systems and systems with relatively lighter loads have not been presented.

In order to handle large scale systems, a new method is presented in (Gemikonakli et al. 2007). Numerical results are computed and validated by simulation but the effects of various numbers of repairmen have not been analysed for large scale multi server systems.

Several analytical methods have been reported for the performability evaluation of multi-server systems. In (Chakka and Mitrani 1994; Chakka et al. 2002) multi-server systems are considered with various repair disciplines (e.g. first broken first repaired, round robin etc.). Markov models are presented for heterogeneous multi-server systems with various repair strategies. Also in (Chakka and Mitrani 1992; Ever et al. 2007) numerical results presented using the spectral expansion method for various multi-server systems with breakdowns and repairs. However, the solutions used in these publications are applicable to small and medium size systems only. Larger networks give rise to state space explosion problem and most analytical solution techniques become inadequate. Mitrani's dominant eigenvalue approach (Mitrani 2005) works well especially for loaded networks due to the fact that dominant eigenvalue better represents loaded networks. This is an improvement of the Spectral Expansion method (Chakka and Mitrani 1992), however, state space is still limited with matrix sizes. Also the dominant eigen value approach may not give good approximations for systems which are not heavily loaded (Mitrani 2005). Furthermore, these works do not address the optimization of the number of repairmen for large systems.

Quasi Birth and Death processes (QBD) (Ciardo and Smirni 1999; Hung and Do 2001; Wallace 1969) have a wide range of applications in Queuing systems. A QBD process can be described as a two-dimensional Markov chain where transitions are only possible between adjacent states of a given model. The popularity of the QBD processes leads to the development of various numerical procedures for their steady-state analysis (Akar and Sohraby 1997; Bini and Meini 1996; Chakka and Mitrani 1992; Haverkort and Ost 1997; Hung and Do 2001; Krieger et al. 1998; Latouch and Ramaswami 1993; Naumov et al. 1996; Neuts 1981). These approaches have received considerable attention. In (Haverkort and Ost 1997) two such approaches are compared. An important difficulty associated with these methods is the state space explosion problem (Haverkort and van Moorsel 1995), which, in case of clusters, limits the analysis and performability evaluation of the number of processors working in parallel. Events such as break-downs, and rebooting/reconfiguration further increase this problem. In today's world, several hundred processors are likely to run in parallel.

Two-dimensional models with multiple components are effectively used for the modelling of queuing systems with multiple queues and/or multiple servers. The functional equations arising in the analysis of such processes usually present significant analytical difficulties (Boxma et al. 1994). These numerical difficulties are frequently caused by large number of steady states. In other words, difficulty caused by the rapid increase in the size of the state space of the underlying Markov process gives rise to the state space explosion problem.

When two dimensional models are considered, the cause of large number of state spaces can be either large (or infinite) number of jobs or large number of operative states (e.g. number of servers). For multi-dimensional models, where one component is finite there are good analytic-algorithmic methods, such as Matrix-geometric solution (Neuts 1981), and Spectral Expansion method (Chakka and Mitrani 1992). These methods can be used to solve the state explosion problem caused by large or infinite number of jobs (El-Rayes et al. 1999). Although systems with unbounded queuing capacities can be handled by Spectral Expansion and Matrix-geometric solution methods, as the number of operative states increases they become computationally expensive. The numerical complexity of the solutions depends on the number of operative states (Mitrani 2005). That number determines the size of the matrix R used in Matrix-geometric method, and the number of eigenvalues and eigenvectors involved in Spectral Expansion method. In both solution techniques the size of the matrices used depends on the number of operative states of the system considered. As the size of the matrices increases the computational requirements increase significantly. Because of the large size, ill-conditioned (Mitrani 2005) matrices numerical problems occur. Also, the numerical stability of the solution gets affected especially when heavily loaded systems are considered.

In this paper, a simple and flexible approximate technique capable of overcoming state space explosion problem in computing the performability of large Markov models is presented. The technique is used to analyse the effects of the number of repairmen working on the repair of failed processors, on overall performability of large scale multi server systems. For validation of the approximate solution method employed, simulation has been used. It is shown that, this approach can handle large state space. The proposed method allows analysing the breakdown repair behaviour of large multi server systems. A Poisson stream of arrivals together with exponentially distributed service times, time between failures, and repair times are assumed. An optimisation study is presented together with numerical results showing the relationship of the number of servers and number of repairmen for optimum performance.

2. APPROXIMATE SOLUTION FOR THE STEADY STATE PROBABILITIES OF LARGE MARKOV MODELS

The proposed approximate solution method uses an iterative technique to calculate the steady state probabilities of a two-dimensional Markov chain. Consider a K-server, homogeneous system with a queuing capacity L, number of repairmen R, mean arrival rate σ, mean service rate μ, mean break-down rate ξ, and mean repair rate η. On the lattice, while one-step downward transition rates (j to j-1) can be represented in terms of μ, i, and j, one step upward transition rates (j to $j+1$) depend on σ. The lateral transitions from i to i+1, and i to i-1 can be expressed in terms of $(K-i)$, R, η, and $i\xi$ respectively. Consider a continuous time, two-dimensional Markov process on a finite lattice strip. The Markov Process can be defined as $X=\{I_n, J_n; n=0, 1, \ldots\}$ with a state space of $(\{0, 1, 2, \ldots, K\}\mathrm{x}\{0, 1, 2, \ldots, L\})$ where, for a multiprocessor system, K and L represent the number of processors and the queuing capacity respectively. Then, i=0, 1, 2, …, K, and j=0, 1, 2, …, L can be used to represent all possible states, (i,j), on the lattice strip. Hence, the steady state probabilities can be denoted as $p_{i,j}$. Figure 1 shows the states of such a system, where, $r_i = \min(R, K-i)$ and $k_i = \min(j, i)$.

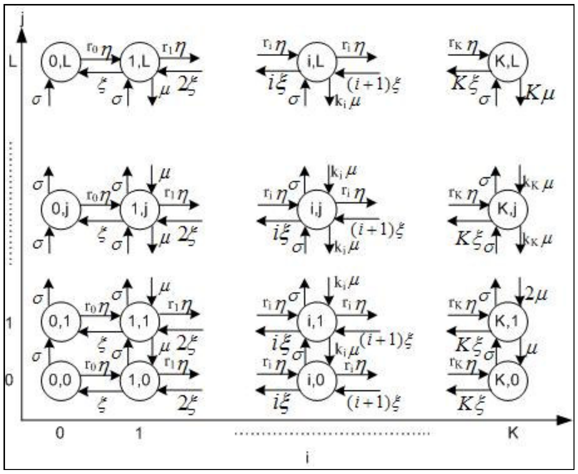


Figure 1: The states of the system under study

Let's define *column* vectors $\mathbf{v}_i$ as follows:

$$\mathbf{v}_i = \{p_{i,0}, p_{i,1}, p_{i,2}, \ldots, p_{i,L}\} \text{ for all } i=0,1,.., K \tag{1}$$

where i represents the number of operative servers, and $p_{i,j}$ are the steady state probabilities. For each i, $\mathbf{v}_i$ can be calculated using product form formulae. From Figure 1, for each column (i.e. i=0, 1, …, K), a set of balance equations can be obtained. An example of these balance equations is given in Eq. (2) for (i, j), where, $0 < i < K$, and $0 < j < L$.

$$\begin{aligned} &\sigma p_{i,j-1} + r_{i-1}\eta p_{i-1,j} + (i+1)\xi p_{i+1,j} + k_{i,j+1}\mu p_{i,j+1} \\ &= (k_{i,j}\mu + \sigma + i\xi + r_i\eta)\, p_{i,j} \end{aligned} \tag{2}$$

By using a set of balance equations which can be derived from (2), together with $\sum_{i=0}^{K}\sum_{j=0}^{L} p_{i,j} = 1$, (3) is obtained.

$$p_{i,j} = \begin{cases} \left(\rho^j p_{i,0}\right)/\, j!, & 0 < j \le i \\ \left(\rho^j p_{i,0}\right)/\left(i!\, i^{j-i}\right), & i < j \le L \end{cases} \tag{3}$$

where, $\rho = \sigma/\mu$.

Next, let's introduce S_i representing the sum of all state probabilities for each operative state i; $S_i = \sum_{j=0}^{L} p_{i,j}$. Since each of the operative states are reached through server break-downs or repairs (horizontal transitions), the following general equation can be obtained:

$$r_{i-1}\eta S_{i-1} + (i+1)\xi S_{i+1} = (r_i\eta + i\xi) S_i .$$

Hence Eq. (4) can be obtained as following:

$$S_i = S_0 \frac{(\eta/\xi)^i}{i!} \prod_{m=0}^{i-1} r_m \tag{4}$$

where

$$S_0 = \left[1 + \sum_{k=1}^{K} \left(\frac{(\eta/\xi)^k}{k!} \prod_{m=0}^{k-1} r_m \right) \right]^{-1} \tag{5}$$

Furthermore, Eq. (5) can be used to calculate the probability that all servers are idle for a given i. For this, first using (3), it can be shown that;

$$S_i = \left(\sum_{j=0}^{i} \frac{\rho^j}{j!} + \sum_{j=i+1}^{L} \frac{\rho^j}{i!\, i^{j-i}} \right) p_{i,0} \text{ , and hence,}$$

$$p_{i,0} = S_i \left(\sum_{l=0}^{i} \left(\frac{\rho^l}{l!} \right) + \frac{i^{L-i}\rho^{i+1} - \rho^{L+1}}{i!\, i^{L-i}(i-\rho)} \right)^{-1} \tag{6}$$

Remaining state probabilities can then be calculated using (3). Furthermore, $p_{0,0}$ can be obtained as follows:

$$p_{0,0} = \xi(R\eta + \sigma)^{-1} p_{1,0} \tag{7}$$

(3)-(7) define all approximate steady state probabilities, $\mathbf{v}_i$, for i operative servers. These equations can now be used in calculating approximate $p_{i,j}$. These probabilities are approximate because lateral transitions have not been taken into account yet. Once the approximate steady state probabilities are calculated, the balance equations given in (7) – (15) can be used to calculate the steady state probabilities more accurately. It is assumed that the number of repairmen is less then number of servers, and in case of breakdowns each server is repaired by a single repairman. Equations (8) – (15) can be given as follows:

$$p_{0,j} = \left(\xi p_{1,j} + \sigma p_{0,j-1}\right)\left(R\eta + \sigma\right)^{-1}, \tag{8}$$
for all j, $0<j<L$

$$p_{0,L} = (R\eta)^{-1}\left(\xi p_{1,L} + \sigma p_{0,j-1}\right) \tag{9}$$

$$p_{i,0} = \frac{\left((i+1)\xi p_{i+1,0} + \min(K-i+1,R)\eta p_{i-1,0} + \mu p_{i,1}\right)}{i\xi + \sigma + \min(K-i,R)\eta} \tag{10}$$
for all i, $0<i<K$

$$p_{i,j} = \frac{\begin{pmatrix} (i+1)\xi p_{i+1,j} + \sigma p_{i,j-1} + \\ \min(K-i+1,R)\eta p_{i-1,j} + \min(i,j+1)\mu p_{i,j+1} \end{pmatrix}}{\min(i,j)\mu + i\xi + \sigma + \min(K-i,R)\eta} \tag{11}$$
for all i, j, $0<j<L$, $0<i<K$

$$p_{i,L} = \frac{\left((i+1)\xi p_{i+1,L} + \sigma p_{i,L-1} + \min(K-i+1,R)\eta p_{i-1,L}\right)}{\min(i,L)\mu + i\xi + \min(K-i,R)\eta} \tag{12}$$
for all i, $0<i<K$

$$p_{K,0} = \left(\eta p_{K-1,0} + \mu p_{K,1}\right)\left(K\xi + \sigma\right)^{-1} \tag{13}$$

$$p_{K,j} = \frac{\left(\sigma p_{K,j-1} + \eta p_{K-1,j} + \min(K,j+1)\mu p_{K,j+1}\right)}{\min(K,j)\mu + K\xi + \sigma} \tag{14}$$
for all j, $0<j<L$

$$p_{K,L} = \frac{\left(\sigma p_{K,L-1} + \eta p_{K-1,L}\right)}{K(\mu + \xi)} \tag{15}$$

Since the steady state probabilities for state (i, j) are initially calculated independent of states (i-1, j) and (i+1, j), it is important to use a technique to compensate for the unaccounted effects of the latter two states on state (i, j). This can be achieved through the use of the balance equations (7) – (15) together with an iterative process. We have K+1 vectors, if $t_{x,,y}$ represents transitions from vector x to vector y where $x \neq y$, then, the transitions can be summarized as follows:

$$[\mathbf{v}_0]\overset{\rightarrow t_{0,1} \rightarrow}{\underset{\leftarrow t_{1,0} \leftarrow}{}}[\mathbf{v}_1]..[\mathbf{v}_{i-1}]\overset{\rightarrow t_{i-1,i} \rightarrow}{\underset{\leftarrow t_{i,i-1} \leftarrow}{}}[\mathbf{v}_i]..[\mathbf{v}_{K-1}]\overset{\rightarrow t_{K-1,K} \rightarrow}{\underset{\leftarrow t_{K,K-1} \leftarrow}{}}[\mathbf{v}_K]$$

Then, an iterative procedure can be followed to accurately calculate $p_{i,j}$. The procedure can be given as follows:

(i). First, $\mathbf{v}_i$ are calculated for i = 0, 1, 2, …, K using (1) , (3), (4), (5), and (6).
(ii). Knowing approximate $p_{i,j}$, the balance equations given in (7) - (15) are used to calculate the correct steady state probabilities.
(iii). The sum of all probabilities is calculated for the queuing system considered.
(iv). Steps (ii) and (iii) are repeated until the sum of probabilities converges to one. Once the correct state probabilities are obtained, various performability measures can be calculated.

The iterative method presented in this study is much faster than simulation and more importantly, unlike most analytical techniques, it can handle large numbers of servers working in parallel without giving rise to state space explosion problem for most practical systems. Furthermore, the method provides accurate results for both heavily loaded networks and networks with relatively lighter loads.

3. NUMERICAL RESULTS AND DISCUSSIONS

Numerical results are presented in this section in order to show the effects of the number of repairmen on systems performability. For all computations parameters are taken as, μ=2, ξ=0.01, η=0.5, and L=1000.

Figures 2-7 show the MQL performance of K-server systems as a function of σ, R and K. It is clear from Figure 2 that when only one repairman is present, increasing the number of servers will not result in an increase in MQL performance. As the number of repairmen is increased, the effects of the number of processors become more significant. Considering that, failure rate can be expressed as $i\xi$, while mean repair rate is $\min(K\text{-}i,\ R)\eta$, the relationship between these two parameters is important. For system efficiency, $R\eta > K\xi$ should be satisfied. As an example, for R=2, and K=64, we obtain 1 > 0.64. Results also show that there is an upper limit for R as well. For a 64-server system, increasing R from one to two significantly improves systems MQL performance and R>2 has no significant effect on system performance. Figure 8 shows the mean queue length of a 512-server system as a function of R and σ. Here, R plays a significant role, the significance of R will only diminish when the network is extremely loaded or R is large (e.g. R >10).

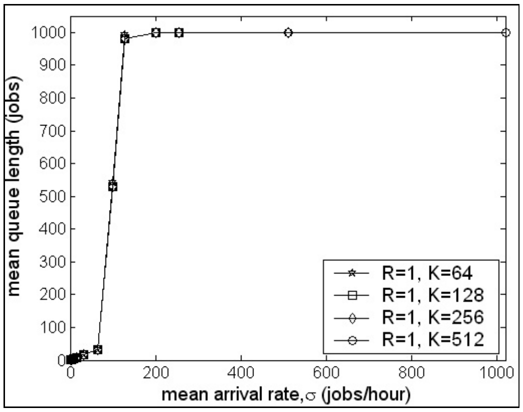


Figure 2: MQL Using a Single Repairman

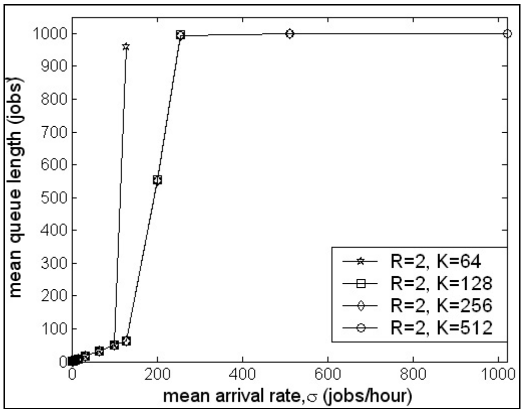


Figure 3: MQL for R=2

Further computations have been carried out to demonstrate the effects of R on system performance for K=512.

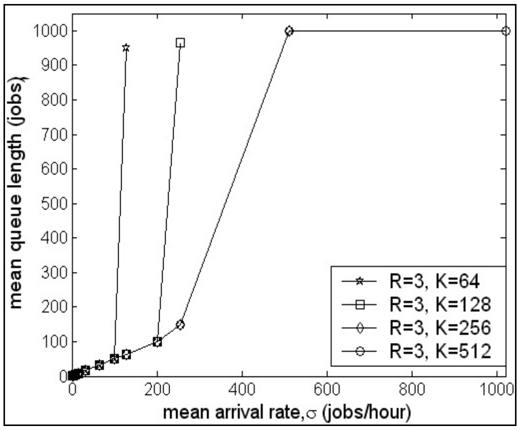


Figure 4: MQL for R=3

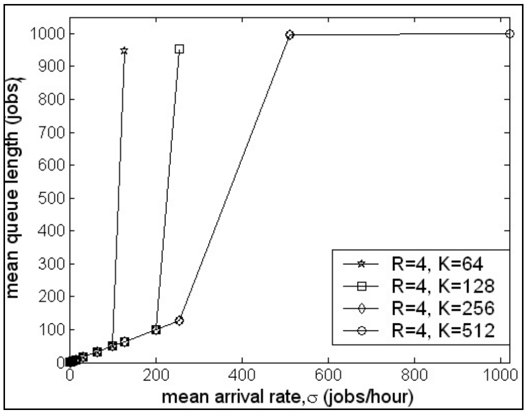


Figure 5: MQL for *R*=4

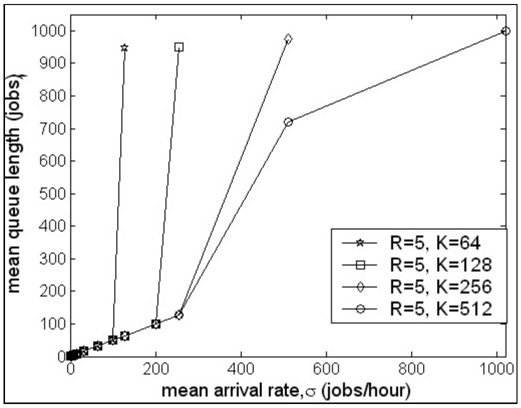


Figure 6: MQL for *R*=5

Figure 9 shows the effect of *R* on the throughput of the system, while Figure 10 shows the effect of *R* on the mean response time of such a system. Results clearly show that there is a threshold for *R* which depends on not only *K* but also σ. Increasing *R* beyond that threshold does not have much significance on mean system response time. However, when the system throughput is considered, the significance of *R* increases as the system's load increases. Again, large *R* does not seem to have much significance on system performability.

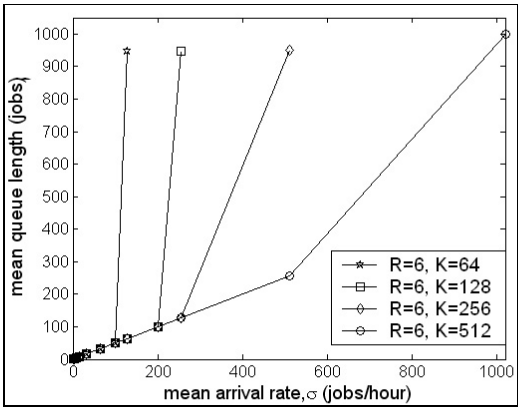


Figure 7: MQL for *R*=6

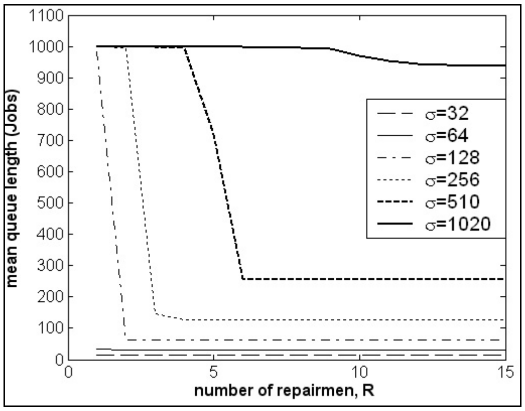


Figure 8: MQL as a Function of *R* and σ, for *K*=512

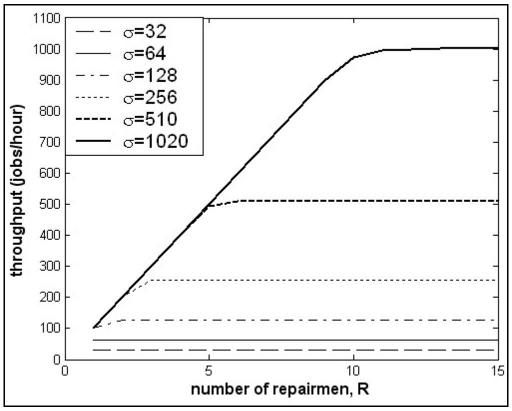


Figure 9: Throughput for *K*=512 and Various *R*

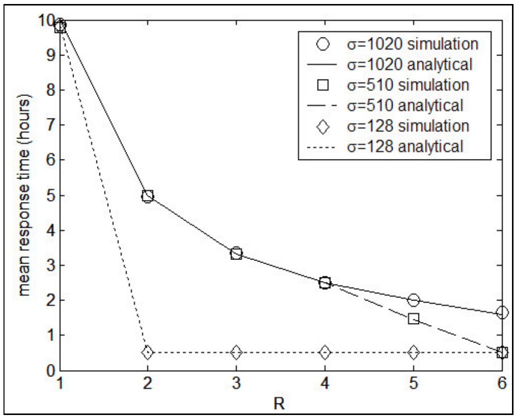


Figure 10: Mean Response Time as a function of *R*, for *K*=512

4. MODEL VALIDATION

In this section, we examine the validity of the approximate analytical model proposed in this paper by computer simulations. For this purpose, we have developed an event driven simulation system in the C++ programming language. The simulator developed simulates the actual multi-processor system under consideration. In order to ensure that the developed simulator correctly simulates the intended system, several tests have been carried out. Some of these tests are as follows:

- Unit tests and code walkthroughs have been conducted during the software development.
- Job arrival, job service, processor failure, and processor repair events and the system state for several simulation runs are traced for consistency checking purposes.
- For various multi-server system configurations with different parameters, the operative state probabilities obtained from the simulation are compared to the values obtained from closed form expressions (equations 4 and 5).
- M/M/K/L queuing systems are simulated (by disabling failures) with different K, L, arrival rate, and service rate parameters and the results obtained from the simulation are compared to the values obtained from analytical solutions (using well-known closed form expressions - see equation 3).

For the validation of the proposed method, we have carried out extensive simulation experiments and reached very close performance conclusions for various scenarios. For simulations we used a 5% confidence interval with a confidence level of 95%.

Figures 10-11 and Tables 1-4 present MQL results obtained using both the proposed analytical method and simulation.

Table 1: A Comparison of MQL Results From the Proposed Analytical Approach and Simulation for K=512

Analytical						
σ	R=1	R=2	R=3	R=4	R=5	R=6
64	32.947	32	32	32	32	32
128	982.425	64.030	64	64	64	64
256	999.345	994.127	147.429	128	128	128
510	999.755	999.384	998.537	995.915	719.123	257.359
1020	999.891	999.755	999.582	999.351	999.031	998
Simulation						
σ	R=1	R=2	R=3	R=4	R=5	R=6
64	32.234	31.971	32.007	32.017	32.035	32.0078
128	983.018	64.032	63.999	64.011	63.989	64.0056
256	997.652	991.711	150.787	128.107	127.925	128.022
510	999.011	998.529	997.493	994.611	729.035	257.7967
1020	999.851	999.702	999.55	999.3	998.986	998.4893

Table 2: A Comparison of MQL Results From the Proposed Analytical Approach and Simulation for K=256

Analytical					
σ	R=1	R=2	R=3	R=4	R=5
64	32.9465	32	32	32	32
128	982.425	64.0302	64	64	64
256	999.345	994.127	147.429	128	128
510	999.755	999.348	998.537	995.944	973.966
Simulation					
σ	R=1	R=2	R=3	R=4	R=5
64	33.0037	31.9974	32.0007	32.0014	31.9592
128	977.6245	64.0356	64.0141	63.9736	63.9754
256	998.3492	991.6139	146.1522	128.0276	128.0276
510	999.6659	999.2278	998.3227	995.7597	973.394

Table 3: A Comparison of MQL Results From the Proposed Analytical Approach and Simulation for K=128

Analytical				
σ	R=1	R=2	R=3	R=4
64	32.9465	32	32	32
128	982.425	64.0303	64	64
200	998.956	553.004	100.232	100.051
256	999.345	995.419	965.345	952.776
Simulation				
σ	R=1	R=2	R=3	R=4
64	33.3109	32.0099	32.0271	32.0043
128	978.3478	64.0106	63.9932	64.0448
200	998.3102	551.8926	100.1956	100.027
256	999.1635	995.2569	964.1577	953.292

Table 4: A Comparison of MQL Results From the Proposed Analytical Approach and Simulation for K=64

Analytical				
σ	R=1	R=2	R=3	R=4
64	32.9694	32	32	32
100	542.46	50.416	50.2514	50.2358
128	994.453	959.3	949.691	947.646
Simulation				
σ	R=1	R=2	R=3	R=4
64	32.3929	32	32.035	32.0099
100	548.9022	50.5001	50.2368	50.2445
128	993.862	958.5472	951.5519	945.927

The results presented in Tables 1-4 clearly show that the discrepancy of analytical results and simulation results are less than 5%.

In addition to MQL, throughput and response times have also been computed using the analytical approach and compared to simulation results. Response time results are shown in Figure 10.

In (Gemikonakli et al. 2007) computational efficiencies of the proposed system and the Spectral Expansion method are compared. In that study *K*=49 has been considered; this is due to state space limitations of the Spectral Expansion method. Computational performances of the two methods are close for a range of utilisation values, however, for loaded networks, the proposed technique is slower than the Spectral Expansion method, but still much faster than simulation.

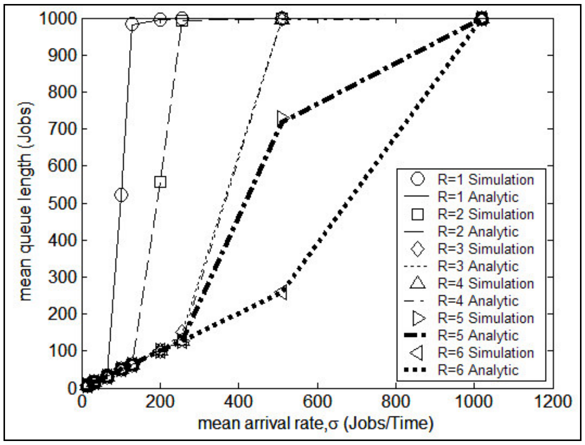


Figure 11: MQL as a function of R and σ and K=512

5. CONCLUSION AND FUTURE DEVELOPMENTS

In this paper, an analytical approach is used to calculate the optimum number of repairmen for large scale, fault tolerant, homogeneous, multi-server systems, to achieve the best performability possible. Inter-arrival times, service times, mean time between failures, and repair times have been assumed to be exponentially distributed. In order to avoid the state space explosion problem inherent to most analytical approaches, an approximate solution technique is proposed for the performance evaluation of such systems. Results obtained using this approach has been confirmed to be accurate using simulation results. The proposed technique has been used to compute the mean queue length, mean response time and throughput of various systems for different numbers of repairmen. The approach lends itself as a powerful technique in evaluating and optimising the performance of large scale networks under various scenarios. The case studies concerned show the strong relationship between number of processors and number of repairmen for operating a large scale multi-server system efficiently. Some threshold values have been obtained for specific case studies. For small and medium scale systems $R\eta/K\xi >> 1$, and hence specific relationship between R and K is not that evident. This case further highlights the importance of the approach used. The proposed method is a flexible one and can be extended to the case of heterogeneous systems, systems with rebooting and reconfiguration delays, highly available systems, Beowulf systems and many other practical, fault tolerant multi-server systems.

REFERENCES

Akar, N., Sohraby, K. 1997. Finite and Infinite QBD Chains: A Simple and Unifying Algorithmic Approach. *Proceedings of IEEE INFOCOM*, pp. 1105-1113. April 1997, (Kobe, Japan).

Bini, D., Meini, B. 1996. On the solution of a non-linear equation arising in queueing problems, SIAM J. Matrix. *Anal.* Appl. 17, 906-926.

Boxma, O.J., Koole, G.M., Liu, Z. 1994. Queueing-theoretic solution methods for models of parallel distributed systems. *Performance Evaluation of Parallel and Distributed Systems, eds.*, pp. 1-24. (CWI Tract 105, Amsterdam)

Chakka, R., Mitrani, I. 1992. A numerical solution method for multiprocessor systems with general breakdowns and repairs. *Proceedings of the 6th International Conference Modelling Techniques and Tools*, pp. 289-304. September 1992, (Edinburgh, UK)

Chakka, R., Mitrani, I. 1994. Heterogeneous Multiprocessor Systems with Breakdowns: Performance and Optimal Repair Strategies. *Theoretical Computer Science 125(1)*, pp. 91-109.

Chakka, R., Gemikonakli, O., Basappa, P. 2002. Modelling multiserver systems with time or operation dependent breakdowns, alternate repair strategies, reconfiguration and rebooting delays. *Proceeding of the SPECTS 2002*, pp. 266-277, July 2002 (San Diego, CA, USA)

Ciardo, G., Smirni, E. 1999. ETAQA: An Efficient Technique for the Analysis of QBD-processes by Aggregation. *Performance Evaluation 36-37*, pp. 71-93.

El-Rayes, A., Kwiatkowska, M., Norman, G. 1999. Solving Infinite Stochastic Process Algebra Models Through Matrix-Geometric Methods. *Proceeding of the 7th Process Algebras and Performance Modelling Workshop, PAPM'99*, pp. 41-62. September 1999, (University of Zaragoza),

Gemikonakli, O., Sanei, H., Ever, E. 2007. Approximate Solution for the Performability of Markovian Queuing Networks with a Large Number of Servers. *Proceeding of the 5th International Workshop on Signal Processing for Wireless Communication, SPWC 2007*. June **2007,** (King's College, University of London).

Haverkort, B.R., Ost, A. 1997. Steady-State Analysis of Infinite Stochastic Petri Nets: Comparing the Spectral Expansion and the Matrix-Geometric Method. *Proceeding of the Seventh International Workshop on Petri Nets and Performance Models*, pp. 36-45.June 1997 (Saint Malo, France).

Haverkort, B.R., van Moorsel, A.P.A. 1995. Using the Probabilistic Evaluation Tool for the Analytical Solution of Large Markov Models. *Proceeding of the Sixth International Workshop on Petri Nets and Performance Models (PNPM '95)*, pp. 206-207, October 1995, (Durham, NC).

Hung, T.T., Do, T.V. 2001. Computational Aspects for Steady State Analysis of QBD Processes, Department of Telecommunication. *Periodica, Polytechnica Ser. El. Eng*. Vol.44, No.2, pp. 179-200.

Krieger, U.R., Naumov, V., Wagner, D. 1998. Analysis of a Finite FIFO Buffer in an Advanced Packet-Switched Network. *IEICE Trans. Commun. E81-B*, pp. 937-947.

Latouch, G., Ramaswami, V. 1993. A Logarithmic Reduction Algorithm for quasi birth-death processes. *Journal of Applied Probability, 39*, pp. 650-674.

Mitrani, I. 2005. Approximate solutions for heavily loaded Markov-modulated queues. *Perform. Eval. 62(1-4)*, pp. 117-131.

Naumov, V., Krieger, U.R., Wagner, D. 1996. Analysis of a Multi-Server Delay-Loss System with a General Markovian Arrival Process. *In S.R. Chakravarthy and A.S. Alfa, editors*, Matrix-analytic methods in stochastic models., *volume 183 of Lecture Note in Pure and Applied Mathematics, Marcel Dekker*, September 1996.

Neuts, M.F. 1981. Matrix Geometric Solutions in Stochastic Model. *Johns Hopkins University Press, Baltimore*.

Wallace, V.L. 1969. The Solution of Quasi Birth and Death Processes Arising From Multiple Access Computer Systems. Thesis PhD, University of Michigan.

AUTHORS BIOGRAPHY

ORHAN GEMIKONAKLI obtained his first degree in electrical engineering from Eastern Mediterranean University in Cyprus, in 1984. He then continued his studies at King's College, University of London where he obtained his MSc in Digital Electronics, Computers and Communications and PhD in Telecommunications in 1985 and 1990 respectively. He worked there as a post-doctoral Research Associate for a couple of years before moving in 1990 to Middlesex University, London where he is now the Head of Department, Computer Communications. He is a member of the IEE, a member of the IEEE and a Chartered Engineer. His research interest is in network security and performability modelling of complex systems. His e-mail address is: o.gemikonakli@mdx.ac.uk and his Web-page can be found at http://www.cs.mdx.ac.uk/staffpages/orhan/.

HADI SANEI obtained his BEng in Computer Hardware Engineering from his native country of Iran. Later he obtained his MSc in Computer Networks in July 2006 from Middlesex University in London. He also is a professional member of BCS (British Computer Society). During his master degree he demonstrated his research interest in performance and reliability modelling and simulation of complex systems. Currently he is a Systems Engineer / RAMS analyst at Halcrow Group Ltd (UK offices). Also he is a PhD candidate for a joint research project at UCLSE (UCL Systems Engineering Dept.), University of London and Halcrow Group Ltd. His email address is saneih@halcrow.com.

ENVER EVER obtained his BSc. degree from the Department of Computer Engineering, Eastern Mediterranean University, Cyprus in 2002, his MSc in Computer Networks from Middlesex University, London in 2003, and his PhD degree in Performance and Reliability modelling from Middlesex University, London in 2008. Currently he is a researcher in The University of Bradford's Engineering, Design and Technology Department working on development of various state-of-the-art computer systems in Markov Modelling, Risk, Reliability, and Safety analysis. His research interest is in networks, performance and reliability modelling of complex systems. His e-mail address is eever@bradford.ac.uk.

ALTAN KOCYIGIT obtained BSc, MSc, and PhD degrees from the Middle East Technical University Department of Electrical and Electronics Engineering Department in 1993, 1997, and 2001, respectively. He has been with the Middle East Technical University Informatics Institute since 2002. His research interests include computer networking, software engineering, and parallel/distributed processing. His e-mail address is kocyigit@metu.edu.tr.

A SOLUTION FOR IMPROVED MODELLING EFFICIENCY OF A MULTI-DISCIPLINARY MARINE POWER SYSTEM

Jeroen D. Schuddebeurs(a), Patrick J. Norman(b), Ian M. Elders(c), Stuart J. Galloway(d), Campbell D. Booth(e), Graeme M. Burt(f) , Judith Apsley(g)

(a, b, c, d, e, f) University of Strathclyde, UK
(g) University of Manchester, UK

(a)jeroen.schuddebeurs@eee.strath.ac.uk, (b)pnorman@eee.strath.ac.uk, (c)i.elders@eee.strath.ac.uk,
(d)s.galloway@eee.strath.ac.uk, (e)c.booth@eee.strath.ac.uk, (f)g.burt@eee.strath.ac.uk, (g)j.apsley@manchester.ac.uk

ABSTRACT

Integrated Full Electric Propulsion (IFEP) systems offer increased design flexibility and operational economy by supplying propulsion and service loads from a common electrical system. Predicting the behaviour of IFEP systems through simulation is important in reducing the design risk in a proposed vessel. However the prevalence of power electronics and the potential for interaction between large electrical and mechanical machines introduce significant simulation challenges. This paper presents an integrated IFEP simulation tool, which brings together models from the electrical, mechanical, thermal and hydrodynamic domains, facilitating end-to-end simulation of the behaviour of the propulsion system. This capability enhances the characterisation of modelling interfaces compared to existing tools. The paper discusses the approaches adopted in increasing computational efficiency without unduly compromising the accuracy of simulation results. The model validation process is described, and finally, the paper presents two case studies as an illustration of the phenomena which the model has been used to investigate.

Keywords: all-electric-ship, multi-domain modelling, multi-rate simulation

1. INTRODUCTION

In recent years there has been increasing interest in, and adoption of, Integrated Full Electric Propulsion (IFEP) approaches in both commercial and naval ships. IFEP vessels combine an electric propulsion system with the electric system serving auxiliary and hotel loads into a single common power distribution system (e.g. Hodge and Mattick 1995; Hodge and Mattick 2001; Danan *et al* 2005). This approach is believed to offer a number of benefits, including increased flexibility in the design of vessels, the ability to more closely tailor the engine installation to the vessel's range of duties, and improvements in efficiency, particularly at part load. These benefits may lead to significant financial savings over the lifetime of the ship.

IFEP systems closely integrate a diverse range of electrical and mechanical and hydrodynamic systems through an electrical network with little inertia through which disturbances can propagate very rapidly. Consequently, events in one part of the system, such as an electrical fault, a disturbance at the propeller or simply a sudden change in load, can very quickly have effects on, and provoke responses from, other components. For this reason, improved characterisation of the behaviour of individual IFEP systems is important so that the design of the vessels can be optimised, and to permit effective operation of the ship by the crew, particularly in unusual conditions. Modelling and simulation of IFEP systems, including in particular the inherent interactions between electrical and mechanical systems, is an important element in achieving this objective.

The Advanced Marine Electric Propulsion Systems (AMEPS) consortium, which brings together the expertise of Strathclyde, Manchester and Cranfield Universities, has carried out research to support the development of a high-fidelity simulation tool for electro-mechanical systems, in order to permit the efficient simulation of IFEP systems. The objective of simulations of this type is to obtain a quantified understanding of the interactions between the diverse components through a "whole system" simulation approach (Norman *et al* 2006). Construction of an integrated model within a common simulation environment permits a more complete understanding of system behaviour than could be obtained by analysing each subsystem in isolation.

This paper presents a model of a representative part of an IFEP system which has been constructed by the AMEPS consortium in order to demonstrate the modelling process adopted. Approaches to improve the computational efficiency of the model are outlined, and ways to optimise the balance between efficiency and accuracy are discussed. Model validation is an important consideration in any simulation activity; available methods are reviewed, and the approach adopted in this work is outlined. The paper presents two case studies demonstrating the utility and practical capabilities of the model, thus illustrating the effectiveness of the modeling approach adopted by the consortium.

2. OVERVIEW OF THE AMEPS MODEL

The model developed by the AMEPS consortium (which is shown schematically in Figure 1) brings together three sub-models, each representing a major section of an IFEP system – the electric motor and drive, the power distribution network and the prime mover (in this case, a gas turbine). Each of these models is constructed using the software tools most appropriate for the underlying technology. The motor, drive, propeller and basic hydrodynamics of the ship are modelled using Matlab/Simulink; models of the generator and electrical network, including auxiliary loads and passive filters, are constructed within the SimPowerSystems toolbox of Matlab (Mathworks 2004); and the thermodynamic model of the gas turbine uses FORTRAN code.

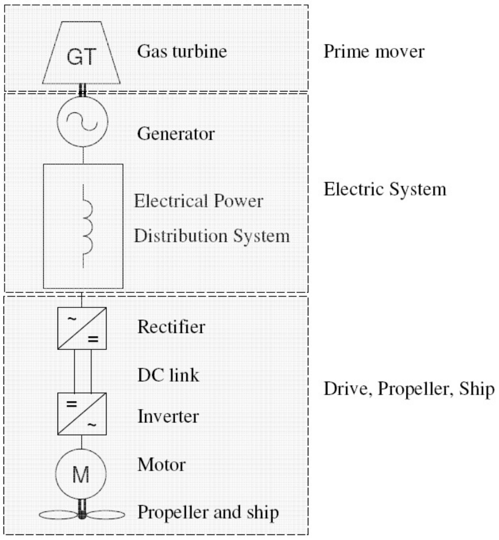


Figure 1: AMEPS Model

The subsystem models were integrated into a single "end-to-end" model within Matlab/Simulink, which permits interconnection of these different modelling approaches. In the following sections, some of the most important challenges of this integration process will be discussed.

3. COMPUTATIONAL EFFICIENCY

A high-fidelity model of a marine IFEP system is necessarily large and complex. Such a model may require significant computational resources in order to carry out simulations, and may consume a significant amount of time to perform each simulation run. Two particular influences on these requirements can be identified (Gole *et al* 1997):

- The level of detail in which typical subsystem models represent the behaviour of equipment may exceed that actually required.
- The need to use short simulation time steps at the same time as simulating events of long duration.

The computational efficiency of the simulation in respect of each of these factors has been optimised as described in the following sections.

3.1. Model Abstraction

When modelling each of the major components within the AMEPS model, and particularly for the faster subsystems such as the power electronic motor drives, care has been taken to model at the minimum acceptable level of fidelity required to fully characterise the phenomena of interest. As well as reducing the computational overhead involved in calculating the state of the model at each simulation time step, this can also enable the use of a larger time step. However, care must also be taken to ensure that this approach does not involve unacceptable approximations or excessive assumptions about overall system behaviour.

For example, the propulsion drive is modelled using a hybrid approach (Apsley *et al* 2007; Gonzalez-Villaseñor, Todd and Barnes 2006) that utilises a detailed diode bridge rectifier model together with an averaged voltage vector inverter model. The use of an averaged rectifier model would also be desirable as this would permit the use of a larger simulation step size for the entire propulsion drive model, further reducing the overall computational burden. However, the switching instants in the diode rectifier are determined by the external circuit conditions on both the AC and DC sides. To predict when these occur, the averaged value model must make assumptions regarding the load current, network voltages and impedances which are not readily applicable to IFEP applications with multi-generator, multi-load power distribution systems. As a result, a detailed diode bridge model, which does not assume fixed network impedances and a balanced supply, has instead been employed.

3.2. Multi-rate simulation

Some components of an IFEP network (such as power electronic converters) experience phenomena which are characterised by very small time constants, of the order of microseconds. As a result, in order to properly characterise these effects and to ensure simulation stability, the time-domain simulation must proceed in very short time steps. For a system of the complexity of an IFEP vessel, this would result in a very large computational burden, which would require significant computing hardware and long simulation times.

It is noticeable, however, that time constants in other parts of the IFEP model are very much longer. Table 1 (Apsley *et al* 2007) shows the wide diversity which may be found in a typical IFEP model.

Table 1. Typical system time constants

Subsystem	Typical Time Constants
Power converter switching	1-5μs
Rotor time constant	50ms – 1s
Propeller run-up time	20s-60s
Ship run-up time	60s-500s

If a common simulation time step were adopted throughout the entire model, computational effort would be unnecessarily expended on high frequency recalculation of the state of elements which only experience slowly varying phenomena. By computing the state of such components less frequently, large efficiencies may be realised.

The AMEPS model implements this concept through a multi-rate simulation approach (e.g. Chen *et al* 2004; Pekarek *et al* 2004; Crosbie *et al* 2007). For each element of the overall model, one of three fixed simulation time steps is selected to satisfy the requirement for adequate characterisation of behaviour without over-simulation. Thus, the gas turbine and propeller-ship models operate with a step size of 1ms, while the main electrical system and propulsion motor models take a 5μs time step to ensure that rapid events such as electrical faults are adequately characterised.

As previously stated, the inverter model adopts an averaged voltage vector behavioural approach to representing the operation of the device. Thus, to capture the averaged switching effects of the converter, a step size of 400μs was selected.

Table 2 shows the practical improvement to the computational efficiency resulting from the use of multi-rate simulation in the AMEPS model in comparison to a single-rate simulation. These results were obtained by averaging the actual elapsed times over multiple simulations of load step events of the indicated "model time" duration.

Table 2: Reductions in simulation time resulting from multi-rate simulation

Simulated event duration (s)	Single rate completion time (s)	Multirate completion time (s)
1	2015	97.4
3	4507	283
5	7970	526

From the results in Table 2 it is observed that the multi-rate simulation is highly beneficial, offering an improvement of up to twenty times in the simulation speed. However, as discussed in the following section, care must be taken to ensure that simulation accuracy is not compromised when simulated values are transported across time-step boundaries.

4. MULTI-RATE SIMULATION VALIDATION

The implementation of a multi-rate simulation has also given rise to new challenges in ensuring that the results do not lose accuracy as a result of transitions between different parts of the model. Two areas of specific interest are addressed in the following subsections.

4.1. Data-transfer Latching

When data is transferred from a part of the model with a short simulation time step into a sub-system with a longer time step, there is a risk that the impact of short-duration, transient phenomena may be inadvertently amplified. To avoid the risk, data transferred must be reflective of the average situation over the longer time step rather than that at the instant of synchronisation (Crosbie *et al* 2007; Pekarek *et al* 2004).

For example, consider a case in which a transient effect of short duration – perhaps a voltage spike lasting for a few time steps – occurs in the behaviour of a component simulated with a short time step, which is adjacent in the model to a component which is simulated with a much longer time step. If the short duration event is taking place at the moment of synchronisation, when data is transferred between the parts of the model, then the slower sub-system may 'latch' on to the transient value. That is, while the transient rapidly dies away in the 'originating' subsystem, its effects are sustained in the 'receiving' subsystem until the next moment of synchronisation.

This phenomenon is illustrated in Figure 2 which shows the transfer of voltage data from the DC-link into the inverter model. The DC-link is in a part of the model which runs at a short time step, whereas the inverter runs at a much longer time step. The graph shows the voltage at the boundary as experienced by the DC-link (grey bars) and the inverter (heavy line). It can be seen from this graph that a short-lived voltage spike at the time of data exchange causes the input to the inverter to 'latch' – that is, to behave as if the transient voltage peak was sustained for a much longer time.

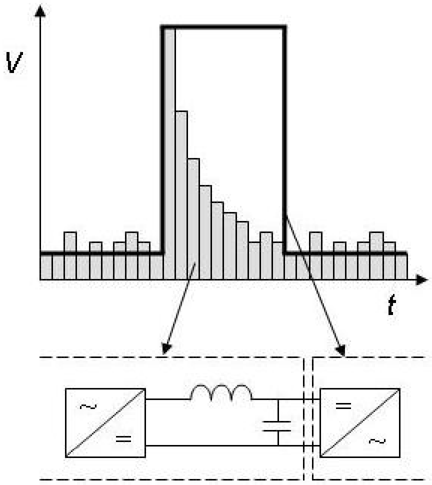


Figure 2: Latching between DC-link and inverter

Given the large time step differences between components in the AMEPS model, this could lead to such transient effects being incorrectly amplified to a significant effect. Some of the events and phenomena which the AMEPS model is intended to investigate, such as electric system faults or sudden load changes in certain parts of the network are likely to lead to problems of this nature, with consequently inaccurate simulation of the behavior of slower-responding components such as the gas turbine.

The AMEPS model uses natural filtering of the data transferred from fast to slow subsystems to reduce the effects of such latching. This approach places the boundaries between different simulation rates in the model at physical boundaries with properties giving inherent resistance to sudden changes in state, such as mechanical inertia or electrical capacitance. Table 3 below lists the fast subsystem to slow subsystem transitions within the AMEPS model and the natural filtering that takes places at each boundary.

Table 3. Natural filtering within the AMEPS model

Fast to slow transition location	Data transferred	Natural filtering aspect
Electrical generator (fast) to gas turbine (slow)	Shaft speed	Shaft and rotor inertia
dc link (fast) to inverter (slow)	Voltage	Inductive and capacitive filter
Propulsion motor (fast) to propeller (slow)	Shaft speed	Motor and propeller inertia

This natural filtering approach is preferable to the addition of explicit filtering or averaging elements at the boundary, since the modelled behavior of and interaction between the adjacent components is not altered. Thus, no artificial sources of error are introduced into the simulation.

Natural filtering has proved satisfactory for all of the simulations conducted to date using the AMEPS model. However it is recognised that where disturbances close to a naturally-filtered boundary are introduced, conflicts may arise between the averaging behavior of the boundary and its interaction with the disturbance. For example, if an electrical fault is simulated in the DC link or inverter, then the interaction between the fault and the inductive and capacitive elements will nullify their filtering effects. Indeed the transient current and voltage effects induced by this interaction may exacerbate the latching problem at this boundary.

In such cases, the introduction of artificial low-pass filtering elements can be considered in order to reduce simulation inaccuracy in the slower subsystem. However, the error introduced by this addition should be balanced against that resulting from the data latching effect to ensure that the lowest possible overall error is achieved.

If it is not possible to balance added filtering against latching to give an acceptable level of overall error, then the simulation time step of the slower subsystem at the boundary can be shortened. This will reduce the error by synchronizing the fast and slow sides of the boundary more frequently, at the cost of longer simulation times.

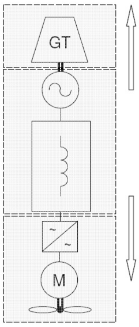
The assessment of the overall effect of these errors on simulation accuracy is not a straightforward task, since it will involve the evaluation of the propagation of the error through other subsystems which are connected to those at the time step boundary. As a result the accuracy of results emanating from those subsystems may be affected; this issue is discussed in more detail in the following section. The existence of closed loop control systems complicates the task further, since the combination of sampling and filtering processes involved may have the effect of compensating for the error, producing an output signal which is close to the 'correct' result without the effect of the error. For example, the propulsion drive controller will attempt to achieve the desired propeller speed with an erroneous DC link voltage, as it would with the 'true' voltage. Although the response of the controller will be different in the two cases, the end result – the propeller speed – may be near-identical. This assumes of course, that the magnitude of the error is not such that it alone drives the controller or the controlled devices into saturation. Therefore, benign controller behaviour as described here cannot be assumed, and careful consideration must be given to the effects of the different controller response on other subsystems.

4.2. Model Error Propagation

In the case discussed above, the controller response prevents errors in the DC link voltage from propagating into the propeller behaviour. However, this will result in the current drawn from the rectifier differing from the "error-free" case. This current variation will disrupt current flows in the remainder of the network, with corresponding disturbance to voltages. Other controllers elsewhere in the system will have their behaviour changed by these variations, which will ultimately alter the response of the generator and the gas turbine. Thus, errors resulting from sampling and filtering in one subsystem within the model can propagate both upstream and downstream in the model – in a similar way to genuine disturbances – and as such, result in inaccuracies in the results generated in other subsystems.

Specifically, the presence of closed loop controllers tends to permit all simulation based errors to propagate back to the field voltage of the generator and to the fuel flow into the gas turbine. These quantities have no further upstream influences and constraints unlike, for example, the gas turbine speed which is influenced by the fuel flow and generator load. Figure 3 below identifies examples of compensation of synchronization errors by controllers, and the wider impact of this compensation.

This paper has already shown that multi-rate simulation is a very effective means of controlling the length of time and level of computing resources required to simulate a marine electrical system. In many cases, it may be vital to the ability to simulate events of realistic duration without resorting to unacceptable model simplification. Nonetheless, as discussed here, it is necessary to take care to understand the implications for model accuracy when applying the approach. Multi-rate simulation also presents challenges for the validation of models, as will be discussed in the next section.



Sampling Error in:	Compensated by:	Impacts on:
Shaft speed	Governor	GT fuel consumption
DC link voltage	AVR and governor	Excitation voltage and GT fuel consumption
Shaft speed	Inverter control	DC link voltage

Figure 3: Error propagation

5. VALIDATION OF AMEPS MODEL

When constructing and using a simulation model, it is important to consider how its validity can be tested and evaluated. In this section of the paper, the selection and application of validation methods to the AMEPS model is discussed.

5.1. Validation

Law (2005) defines validation of a model as a process to determine whether the model is an acceptably accurate representation of the system with in the context the objectives of the study in which it is applied. A model should be designed and developed to address one or more questions which are understood in advance; this also specifies the level of detail required in the model (Law 2005; Sargent 2003).

5.2. Validation Methods

Sargent (2003) describes a number of methods which can be used to assess the validity of a model. Examples include:

- *Face validation*: In this approach, opinions are sought from one or more experts as to the acceptability of the model's construction and/or the behaviour it predicts.
- *Comparison to other models*: The simulation results of the model to be validated are compared with the results of other previously-validated or independently constructed models.
- *Predictive validation*: In this method, simulation results are compared against measurements made in the field obtained by experiment.

Commonly, a number of validation methods would be employed together to provide greater levels of confidence.

5.3. Subsystem Validation of the AMEPS Model

The hybrid propulsion drive subsystem of the AMEPS model has been validated using the *comparison to other models* approach. The hybrid model was compared against an equivalent model constructed using the PLECS piecewise linear element circuit simulation tool (Plexim GmbH 2008). The validation was carried out using time plots of the line-line supply voltages and line currents produced from the hybrid and PLECS models. In both cases, simplified electrical supply and propeller models were used to permit validation in isolation from the remainder of the AMEPS model.

Additionally, *predictive validation* was applied to the motor model. A variety of tests were carried out on a multi-phase induction motor test rig (Apsley *et al* 2007). The test conditions were replicated in the AMEPS motor model and the actual and simulated behaviour compared. Figure 4 shows an example of this comparison, in which the rotational speed of the real and simulated motors are shown when a ramp change in flux current is applied, followed by a step change in torque current. Figure 4 demonstrates the accuracy of the motor model.

A similar approach has been adopted in validating the gas turbine model, for which manufacturer's performance curves have been used as a basis for comparison. *Face validation* of aspects of the dynamic behaviour of the gas turbine was also used.

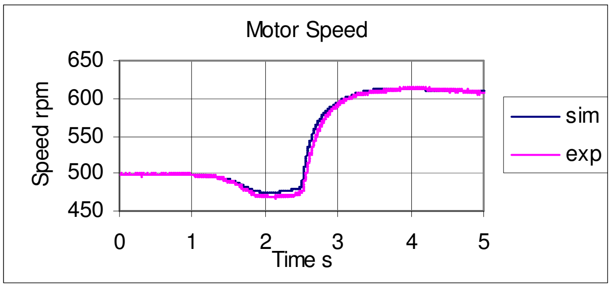


Figure 4: A comparison of experimental and simulation motor speed

The propeller model was validated both by *comparison to other models*, which was useful in validating the implementation of the model, and through *face validation*, in which assistance from domain experts in industry was obtained. This assistance was particularly valuable in validating the underlying mathematical assumptions, and in interpreting the results generated.

Models of electrical network components were mainly validated using *face validation*. In future work, further validation using other approaches, notably the use of hardware-in-the-loop simulation approaches (Palla *et al* 2007) to permit *predictive validation* of component models in the context of a complete network, may be pursued to increase confidence in the validity of the models.

5.4. System Level Validation

As described previously, a number of methods are available for validation of the individual subsystems making up the AMEPS model. Although a set of validated subsystem models might be expected to result in an accurate whole system model, experience has shown that the complexity of the interactions between subsystems, and the risk of incompatible validation assumptions can lead to non-trivial emergent behaviour, suggesting that additional validation would be beneficial.

In principle, the validation methods applied to the subsystems are equally applicable to the validation of the integrated model. In practice, however, a number of difficulties arise. Consider, for example, the question of determining the effect of data latching errors introduced by the multi-rate simulation approach.

Naïvely, it might be assumed that the *comparison to other models* approach could be applied by simply comparing results from the multi-rate model with those from a model entirely simulated at the smallest simulation step size. This would permit straightforward quantification of the overall effect of these errors. However, as discussed above, simulation of many conditions for which the model would need to be validated – for example propeller events – would have impractical requirements of time and computing resources without multi-rate simulation. Therefore, although this validation method has some applicability, other approaches are also needed, particularly as the size and complexity of the integrated model increase.

Predictive validation of the model using field data obtained from IFEP vessels is also attractive. However, detailed data relating to existing vessels is difficult to obtain as a result of confidentiality issues. It is also of limited utility in validating models of vessels which are at the design stage or under construction. Considering that an important benefit of "whole system" simulation is to reduce design risk, this is an important drawback. Construction of a hardware test rig such as the Electric Ship Technology Demonstrator (Mattick *et al* 2005; Danan *et al* 2005) might be an alternative, but is very costly and negates many of the economic benefits of using simulation to de-risk vessels at the design stage. The range of equipment and configuration options which could be investigated is also limited in this approach.

As an alternative, model accreditation (DMSC 2006) was used as a means to assess the validity of the results produced by the integrated AMEPS model. In this process, *face validation* has been carried out by domain experts on simulation results obtained from each subsystem when integrated within the complete model. Although it is recognised that *face validation* is an inherently subjective approach, this is perhaps the best practically achievable solution in the light of the limitations discussed above. It is clear, however, that there is a need to develop a robust framework within which the integrated model can be further validated. This appears particularly important since, as discussed elsewhere in this paper, the level of accuracy in the simulation results may vary according to the scenario being simulated.

6. SIMULATION CASE STUDIES

This section will demonstrate the capabilities of the AMEPS model by presenting two case studies. The first case study assesses the system behaviour after a sudden loss of the propulsion load (caused by a protective trip mechanism within the power electronic motor drive). The second case study assesses the effect of a cyclic propeller loading on the electrical network and prime mover behaviour.

6.1. Model Description

These case studies consider the power distribution network shown in Figure 5, which is similar to one possible operational configuration of the Type 45 Destroyer (Norton and Saxby 2006).

The MV and LV voltage levels are 4160VAC and 440VAC respectively. The gas turbine and propulsion motor are rated at 21MW and 20MW respectively. The MV and LV loads are rated at 2.5MVA and 0.5 MVA respectively with a power factor of 0.85.

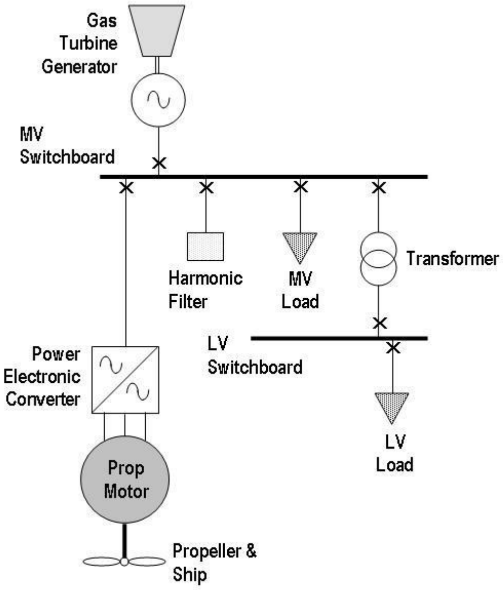


Figure 5: Type 45 single-line diagram

Figure 6 shows the vessel speed control loop. This is a cascade controller where V_s is the vessel speed, ω the AIM speed, Q^* the AIM reference torque and T the propeller thrust. In this case V_s^* is set to 10 m/s, which is kept constant by adjusting Q^* . The propeller is modelled using the Wageningen-B series (Apsley *et al* 2007).

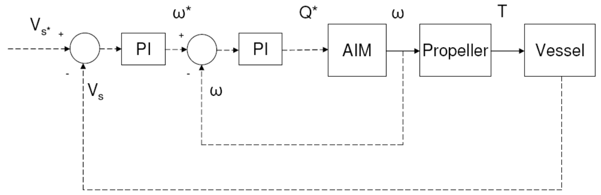


Figure 6: Vessel Speed Control Loop

6.2. Sudden loss of Propulsion Load

This particular scenario investigates the overall system behaviour under severe operating conditions in which the power drawn by the propulsion motor instantaneously drops from the nominal level at cruising speed to zero after 0.5 seconds of simulation time (representing a trip event within the main propulsion drive). Figures 7 to 11 show the simulation results for this scenario.

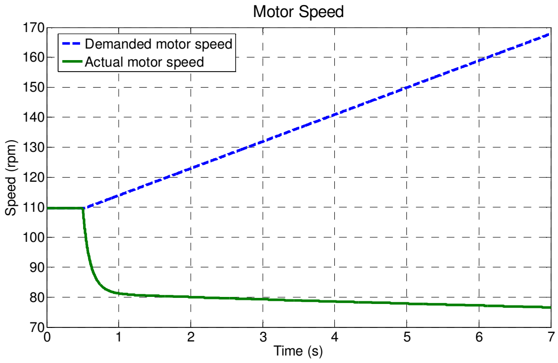


Figure 7: Propulsion Motor Speed

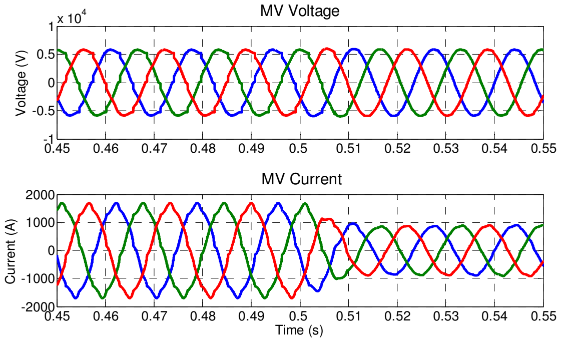


Figure 8: MV Voltage and Generator Current

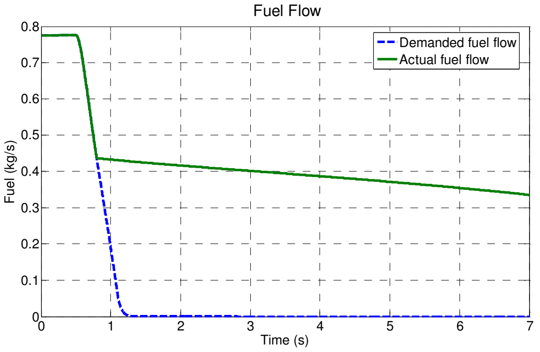


Figure 9: Gas Turbine Fuel Flow

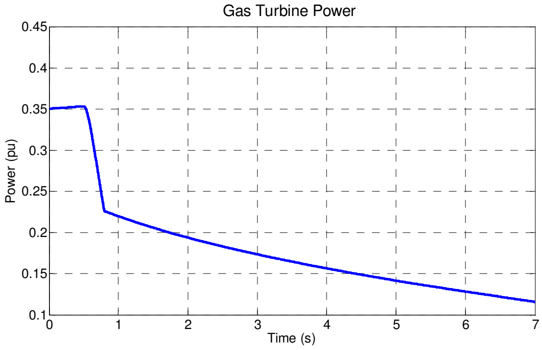


Figure 10: Gas Turbine Power

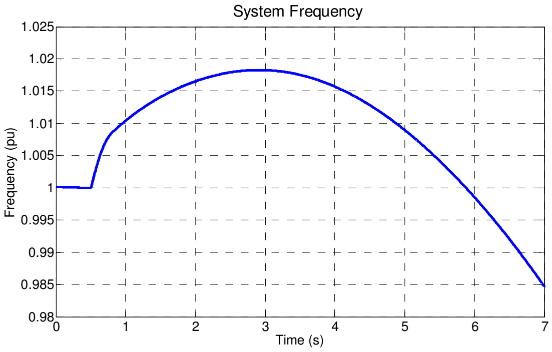


Figure 11: System Frequency

Figure 7 shows a trace of the desired (dotted trace) and actual (solid trace) propulsion motor speed. In this figure the demanded motor speed increases in response to the dwindling vessel speed (not shown). However, the actual motor speed begins to decline following the converter trip as all power to the propulsion drive is lost.

The three-phase MV voltage and current traces (measured at the terminals of the generators) are shown in Figure 8. Note that these traces are shown over a much shorter time frame than the other the parameters presented in order to highlight the waveform distortion evident in these quantities. Prior to the loss of propulsion load, distortion resulting from the operation of the diode bridge rectifier is evident in both traces. Following the loss of propulsion load, however, there is a notable reduction of harmonic content in both traces as the diode bridge ceases to draw any significant power from the main network.

Figure 9 shows the demanded (dotted line) and actual (solid line) gas turbine fuel flow. Immediately after the sudden loss of the propulsion load, there is a surplus of power delivered by the gas turbine. As the gas turbine governor tries to maintain the system frequency at a constant value, it rapidly decreases the fuel flow demand to the minimum level. However, in order to prevent damage to the gas turbine, the rate of change for the actual fuel flow is limited by internal controllers. This limiting action is evident in the plot of actual fuel flow in Figure 9. This in turn causes the power output of the gas turbine (Figure 10) to decrease at a much slower rate than desired by the governor control. As a result, a significant transient in network frequency occurs while the output power of the gas turbine adjusts to the new network loading conditions (Figure 11).

6.2.1. Discussion of Results

This first case study is an excellent illustration of the potential interactions that can take place within IFEP power systems. It clearly illustrates how response of one subsystem to a transient in a separate location can have a substantial impact on the remainder of the power system. Non-linear effects in the gas turbine control have caused exaggerated swings in the network frequency and a particularly poor system response to the original perturbation. Degraded power quality is thus being provided for the remainder of the loads connected to the network. This may have further undesirable consequences, such as nuisance tripping of sensitive loads. An improved control scheme for the gas turbine might be devised, balancing the protection of the prime mover against transients with the effects on the wider IFEP system. This approach may improve the overall system response, although it appears that the initial frequency rise may still be unavoidable, thus preventing a rapid network recovery.

Instead, adopting a coordinated control approach may have a greater impact on mitigating the effects of the propulsion system transient. In this manner, knowing the limitations of the gas turbine in dealing with the loss of load, additional systems within the network (smaller prime movers, electrical loading and energy storage) could be operated more effectively to complement its actions and improve the overall system response to the transient. In this way, a coordinated control approach could provide a substantial increase in functionality over that of isolated control systems.

6.3. Propeller Cyclic Loading

This case study demonstrates the effect of cyclic propeller loading on the entire IFEP system. Such loading can be a result of a vessel cruising in heavy seas where propeller emergence and ‘slamming’ often occurs. In this case study, a simplified cyclic sinusoidal propeller loading profile with a frequency of 0.1Hz and a magnitude of 10% rated thrust has been applied (initiated from 0.5 seconds of simulation time) to illustrate the effects of such loading. Note that this propeller loading profile is in line with the range of realistic values given in (Stewart 2005).

Figures 12 to 16 illustrate the simulation results for this case study.

6.3.1. Discussion of Results

The effect of the cyclic loading can clearly be observed in Figures 12 to 16. In contrast to the previous case study, there is no control saturation present within the gas turbine in this mode of operation, and as such, the actual fuel flow is the same as the demanded fuel flow (Figure 13).

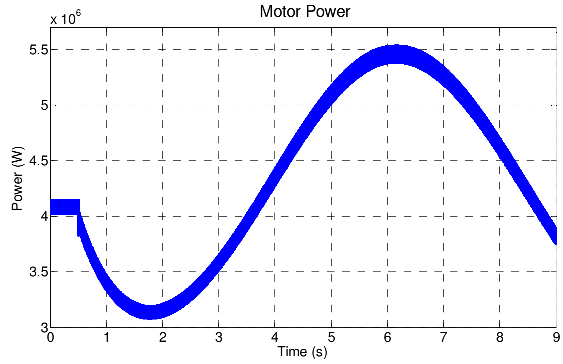


Figure 12: Motor Power

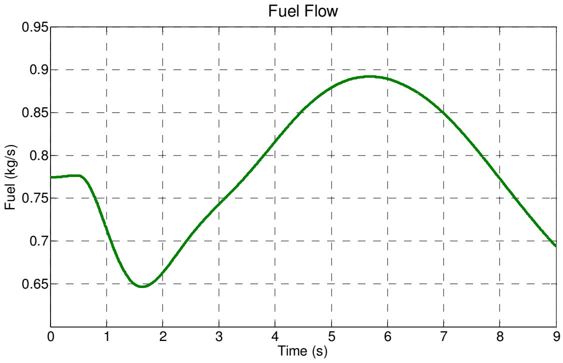


Figure 13: Gas Turbine Fuel Flow

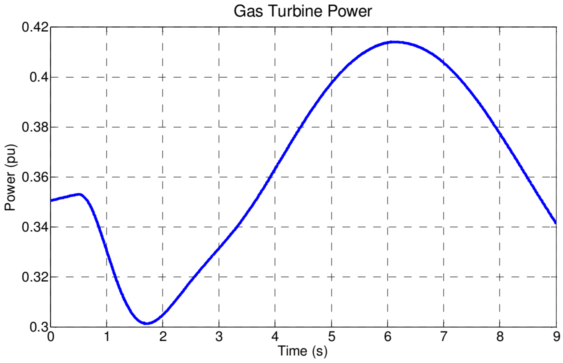


Figure 14: Gas Turbine Power

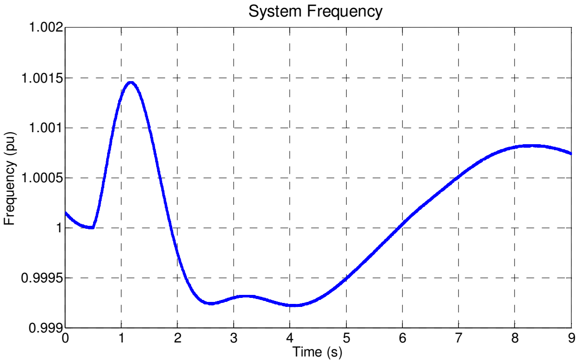


Figure 15: System Frequency

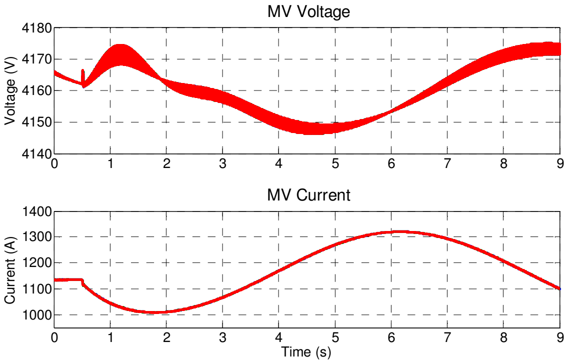


Figure 16: RMS Busbar Voltage and Generator Current

As a result of this behaviour, the response of the gas turbine is sufficient to maintain the network voltage and frequency within acceptable limits despite substantial variation in the magnitude of the network loading.

However, it should be noted that in reality, sea waves do not subject the propulsion systems to a single frequency disturbance but are composed of a range of frequencies (Stewart 2005). The response of the propulsion system to these different disturbance frequencies will vary and this may result in a far greater impact on the prime mover operation and network frequency than that presented here (Elders *et al* 2008). As such, the authors intend to extend the work presented here to consider the impact of a wider range of cyclic loading effects.

7. CONCLUSION

This paper has presented a model developed by the AMEPS consortium to demonstrate the holistic modelling of electrical and mechanical subsystems of an IFEP vessel's propulsion system. Some of the major modelling challenges which have been encountered were identified, and approaches to overcome them discussed. Multi-rate simulation was shown to be a highly effective means of improving the computational efficiency of the model and for reducing the time required for simulations. However, there is a risk that transfer of information between parts of the model simulated at different rates approach may introduce inaccuracies; means for controlling these errors have been discussed in this paper. Validation of simulation models is important in assuring the reliability of the results they produce. A variety of methods have been used to validate the AMEPS model; however only model accreditation can be said to be viable at present as a means of validating the integrated AMEPS model as a whole. Further research into the validation of large and complex models is desirable.

Finally, the case studies presented in the paper demonstrate the capabilities of integrated electrical-mechanical simulation models such as the AMEPS model in assessing the behaviour of IFEP systems when subjected to external events. In an industrial context, this capability will be valuable in, for example, determining compliance with classification society rules.

ACKNOWLEDGMENTS

This paper is part of the EPSRC funded project (GR/S86761/01) on Advanced Marine Electrical Propulsion Systems (AMEPS) between the Universities of Strathclyde, Manchester and Cranfield. The authors would like to thank their colleagues at all three universities for their contribution to this work.

REFERENCES

Apsley J.M., Gonzalez-Villaseñor A., Barnes M., Smith A.C., Williamson S., Schuddebeurs J.D., Norman P.J., Booth C.D., Burt G.M., McDonald J.R., 2007. Propulsion Drive Models for Full Electric Marine Propulsion Systems, *International Electric Machines and Drives Conference (IEMDC07)*, May 5-8, Antalya, Turkey

Chen J., Crow M.L., Chowdhury B.H., Acer L., 2004. An Error Analysis of the Multirate Method for Power System Transient Stability Simulation, *IEEE PES Power Systems Conference and Expositions*, October 10-13, New York, US

Crosbie R.E., Zenor J.J., Bednar R., Word D., Hingorani N.G., 2007. Multi-Rate Simulation Techniques for Electric Ship Design, *Electric Ship Technologies Symposium, Arlington (ESTS2007)*, May 21-23, VA USA

Danan F.G.A., Weston A.V., Longépé B.M.J., 2005. The Electric Ship Technology Demonstrator or How Two MoDs are De-risking the All Electric Warship Concept, *The All-Electric Ship conference (AES2005)*, October 13-14, Versailles, France

Defence Material Standardization Committee (DMSC) 2006. *DEF STAN 61-22 Issue 1 Definition of Modelling Standards – Draft. Marine Electrical Power Systems.*, March 31 .

Elders I.M., Schuddebeurs J.D., Booth C.D., Burt G.M., McDonald R.J., McCarthy J., 2008. Energy Storage Systems as a Mechanism for Improving Power Quality in an IFEP System, *IMarEST 9th International Naval Engineering Conference (INEC2008)*, April 1-3, Hamburg, Germany

Gole A. M., Keri A., Nwankpa C., Gunther E. W., Dommel H. W., Hassan I., Marti J. R., Martinez J. A., Fehrle K. G., Tang L., McGranaghan M. F., Nayak O. B., Ribeiro P. F., Iravani R., Lasseter R., Guidelines for modeling power electronics in electric power engineering applications, *IEEE Transactions on Power Delivery*, Vol. 12, No. 1, pp. 505 - 514, January 1997.

Gonzalez-Villaseñor A., Todd R., Barnes M., 2006. Rectifier and DC-Links simulation models for marine power systems, Technical report University of Manchester, August 9

Hodge C.G., Mattick D.J., 1995. The Electric Warship, *Trans IMarE*, Vol 108, part 2, pp109-125

Hodge C.G., Mattick D.J., 2001. The Electric Warship VI, *Trans IMarE*, Vol 113, Part 2

Law A.M., 2005. How to build valid and credible simulation models, *Proceedings of the 2005 Winter Simulation Conference*, December 4-7, Orlando, US

Mathworks, 2004. *SimPowerSystems™ 4 User's Guide*, Available from: http://www.mathworks.com/access/helpdesk/help/pdf_doc/physmod/powersys/powersys.pdf, [Accessed, May 2008]

Mattick D., Benatmane M., McVea N., Gerrard R., 2005. The Electric Ship Technology Demonstrator: 12 Inches to the foot, *The All-Electric Ship conference (AES2005)*, October 13-14, Versailles, France

Norman P.J., Schuddebeurs J.D., Booth C.D., Galloway S.J., Burt G.M., McDonald J.R., Villaseñor A., Todd R., Apsley J.M., Barnes M., Smith A.C., Williamson S., Mody B., Tsoudis E., Pilidis P., Singh R., 2006. Simulating IFEP Systems, *Marine Engineers Review (MER)*, October, pp26-31

Norton P. T., and Saxby C., 2006. Electric Warship X – The lessons from reality, *Journal of Marine Design and Operations*, No. B10, pp15-26

Palla S., Srivastava A. K., Schulz N.N., 2007. Hardware in the Loop Test for Relay Model Validation, *Electric Ship Technologies Symposium (ESTS2007)*, May 21-23, Arlington VA USA.

Pekarek S.D., Wasynczuk O., Walters E.A., Jatskevich J.V., Lucas C.E., Wu N., Lamm P.T., 2004. An Efficient Multirate Simulation Technique for Power-Electronic-Based Systems, *IEEE Transactions on Power Systems*, Vol. 19, No. 1, pp. 399 – 409.

Plexim GmbH, 2008. *Piecewise Linear Element Circuit Simulation, User Manual, Version 2*, Available from:. http://www.Plexim.com, [Accessed, May 2008]

Sargent R.G., 2003. Verification and Validation of Simulation Models, *Proceedings of the 2003 Winter Simulation Conference*, December 7-10, New Orleans, US

Stewart R.H., 2005. *Introduction to Physical Oceanography*, OceanWorld, Available from: http://oceanworld.tamu.edu/resources/ocng_textbook/chapter16/chapter16_04.htm, [Accessed, May 2008]

AUTHORS BIOGRAPHY

Jeroen Schuddebeurs is a research assistant within the Institute of Energy and Environment at the University of Strathclyde. He received his BSc in Maritime Operations from the Hogeschool Zeeland (The Netherlands) and his MSc in Marine Electrical Power Technology from the University of Newcastle upon Tyne. Currently he is pursuing a PhD in Electronic and Electrical Engineering. His research interests include the dynamic modelling and simulation of more-electric marine power systems.

Patrick Norman is a research fellow within the Rolls-Royce University Technology Centre in Electrical Systems at the University of Strathclyde. His research interests include the modelling and simulation of more-electric marine and aerospace power systems, the analysis of electrical-mechanical interactions within these environments and the fault response of power electronics based DC distribution systems.

Dr Ian Elders is a Senior Research Fellow within the Institute of Energy and Environment at the University of Strathclyde, Glasgow, UK. He received his BEng degree in electronic and electrical engineering and PhD in electrical power engineering from the University of Strathclyde in 1994 and 2002 respectively. His research interests include the control and simulation of marine electrical systems, and the integration of renewable energy into electricity networks.

Dr Stuart Galloway is a lecturer within the Institute for Energy and Environment. He obtained his MSc and PhD degrees in mathematics from the University of Edinburgh in 1994 and 1998 respectively. His research interests include power system optimization, numerical methods and simulation of novel electrical architectures.

Dr Campbell Booth is a lecturer within the Electronic and Electrical Engineering Department at the University of Strathclyde in Glasgow. His research interests lie in the areas of: power system protection; plant condition monitoring and intelligent asset management; applications of intelligent system techniques to power system monitoring, protection and control; knowledge management and decision support systems.

Professor Graeme Burt (M'95) holds a Chair in Power Systems within the Institute for Energy and Environment at the University of Strathclyde. He received his BEng and PhD in Electrical and Electronic Engineering from the University of Strathclyde. He is the Director of the University Technology Centre in Electrical Power Systems sponsored by Rolls-Royce. His current research interests lie in the areas of: network integration of distributed generation; power system modelling and simulation; power system design.

Dr Judith Apsley received her BA degree from the Cambridge University, UK (1986) and her PhD from the University of Surrey, UK (1996). She worked in industry at Westland Helicopters (1982-7), ERA Technology (1987-95) and a small software company (1996-2001), before joining the University of Manchester/UMIST as a research associate, and then lecturer (2008). Her main research interests are the modelling and control of electrical drive systems.

AIRPORT TERMINALS' PERFORMANCE ANALYSIS: A CASE STUDY

Antonio Cimino[(a)], Duilio Curcio[(b)], Giovanni Mirabelli[(c)] Enrico Papoff[(d)]

[(a) (b) (c) (d)] Modeling & Simulation Center - Laboratory of Enterprise Solutions (MSC – LES)
M&S Net Center at Department of Mechanical Engineering
University of Calabria
Via Pietro Bucci, Rende, 87036, ITALY

[(a) (b) (c)] {acimino, dcurcio, g.mirabelli, e.papoff}@unical.it

ABSTRACT

The paper presents a simulation model of an Italian airport terminal, the International Airport of Lamezia Terme in Calabria (Italy). The software tool adopted for the simulation model implementation is Anylogic™ by *XJ Technologies*. After the modeling phase, the simulation model has been validated comparing simulation results with real system results. The performance measure chosen for testing system behavior under different operative scenarios is a mean utilization index of the terminal. The output data of the simulation model are then analyzed by means of ANOVA in order to understand how some critical input parameters (number of security check lines, of passport controls and check-ins) affect system performance.

Keywords: Airport terminal performance, capacity factor, DOE, ANOVA

1. INTRODUCTION

Airport terminals (*ATs*) represent a wide area for applying simulation approaches and techniques in order to test systems' performance under different operative scenarios and to evaluate their structural flexibility, see Verbraeck and Valentin (2002).

In particular, airports have a great relevance because of their key-role as an interface between land and air transportation. Moreover, the importance of airports is related to processes and activities of entities which operate in the same location.

According to Tosic (1992) airport terminals' Modeling and Simulation (*M&S*) has advanced significantly. As reported in Brunetta et al. (1999), models implemented provide enhancements in terms of detail, fidelity and user friendliness. As a consequence, their use as decision support tool in airport terminals' design and management is continuously growing. The airport terminals simulation models can be subdivided into:

- *strategic models* which sacrifice the level of detail for increasing simulation speed and flexibility;
- *tactical models* which are characterized by high detail levels in data management and system representation.

The importance of ATs simulation became more and more important after 11/9 terrorist attacks: in effect, after these attacks security becomes one of the most critical issue of the aviation sector. As reported in Rossiter and Dresner (2004) several security measures have been adopted for avoiding new terrorist actions. According to Glasser et al. (2006), several protective measures have been introduced in order to face each type of problem. Fayez et al. (2008) introduce a decision support tool for supporting airport planners and decision makers in the evaluation of the impact of the changing security regulations and how their application in the airport structure impacts on passengers' service level. The security measures are applied in different fields like security controls and screening procedures for airport passengers, baggage and cabin baggage.

Concerning airport terminals M&S, several studies have been carried out. Sherali et al. (1992) propose new approaches for increasing AT performance and improving its capacity, focusing on increasing the operational use of runways. Snowdon et al. (2000) implement a simulation tool in order to help airlines and airports in using advanced technologies to improve passengers' service level. Other research studies are related to M&S of passengers and baggage flow in airport terminals (Brunetta and Romanin-Jacur, 1999), and innovative solutions for supporting future airport developments (Gatersleben and Van Der Wej, 1999).

The focus of this paper is to present a simulation model of an Italian airport terminal implemented for testing system behavior under different operative scenarios. Section 2 reports the description of the airport terminal being considered; in Section 3, the authors present the simulation model implementation. Section 4 proposes simulation results and analysis while conclusions summarize critical issues and results of the paper.

2. THE AIRPORT TERMINAL DESCRIPTION

The airport terminal analyzed in this research work is the International Airport of Lamezia Terme located in Calabria, (south part of Italy) see Figure 1.

Figure 1: The Airport Terminal of Lamezia Terme

This airport terminal has a key-role in the economic scenery of the region. In fact, it connects Calabria with many national and international cities, through scheduled and charter flights. Built in 1976, during the last years, the International Airport of Lamezia Terme has undergone several changes related to its management and structure, causing as primary consequence the increase of the passengers' number per year. The 2006 represents a record-year for the airport because of the increase in the commercial aviation passengers and flights' number due to quality and functionality of the structure and its services as well as the efficiency of the airport/town connections.

3. THE AIRPORT SIMULATION MODEL

According to Jim and Chang (1998) there is an imbalance in passenger terminal, airfield and airspace planning at airport terminals. Making considerations on an airport structure, it is possible to detect three different areas:

- the airspace, the section of the airport used by different aircrafts in flight;
- the airfield, the airport area used for aircrafts ground movements;
- the passengers' terminal, used by passengers, staff and crews.

According to Curcio et al. (2007), an airport simulation model should help users in improving the airport terminal management. In effect Hafizogullari et al. (2002) explain that M&S is the only way to represent large-scale problems like those that characterize airport terminals. In this case, the M&S approach is introduced because of its capability to capture complex relationships, scalability and interdependencies among entities of the system analyzed. In this research work, the authors develop a simulation model for evaluating airport terminal performance. The next section explains the simulation model architecture.

3.1. The model architecture

The simulation model reproduces all the most important processes and operations of the airport terminal, related to:

- passengers;
- baggages;
- aicrafts' flow.

The software tool adopted for the model implementation is the commercial package Anylogic™ by *XJ Technologies*. In particular, for reproducing each process and for increasing model flexibility, different classes have been implemented by using library provided by the software. A deeper description of classes implemented is reported in Curcio et al. (2007). Figure 1 displays the structure diagram of the simulation model.

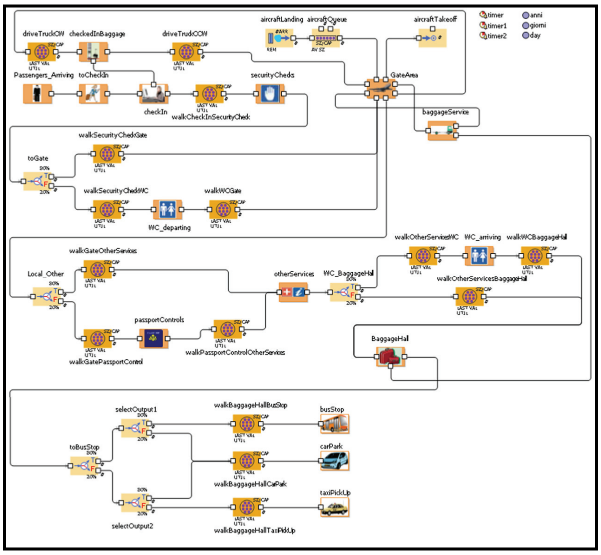


Figure 2: The Structure Diagram of the Model

The structure diagram contains different classes (represented by rectangles) each of them reproducing processes and activities that characterize the real system. More in detail, the approach adopted by authors is the composition approach. As reported in Klein (2000), this approach is based on the segmentation of the system architecture in several functional, geographical components separately implemented. In fact, in this research work the authors implement different functional components, reported as follows.

- *GatesArea*: this is the main class of the simulation model. In this class all the processes and operations related to passengers' departures and arrivals and to aircraft boarding and getting off operations have been implemented.
- *CheckInArea*: this class is implemented in order to reproduce all the operations related to passengers' and baggage check in;

- *SecurityChecksArea*: this class is implemented in order to reproduce all the activities that characterize the security control points;
- *PassportControlsArea*: in this class the operations related to the passport control process are implemented;
- *BaggageArea*, reproduce all the operations related to the baggage hall and the operations related to baggage handling through the terminal;
- *PassengersArrivalArea*: this class reproduces all the operations related to passengers generation ;
- *GeneralServicesArea* which reproduces all the other processes that take place in the terminal;
- *ExitOperationsArea*: this class is introduced for reproducing all the processes related to bus transportation, taxi services and car park.

3.1.1. The SecurityChecks Area

This class reproduces all the operations related to passengers' and baggages' security controls. Figure 3 shows the SecurityChecks Area structure diagram.

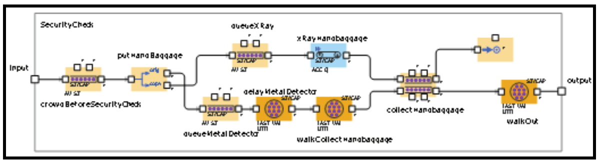


Figure 3: The Security Checks Structure Diagram

The logical sequence of all the activities reproduced in this section by using a block diagram is as follows (see Fig.4).

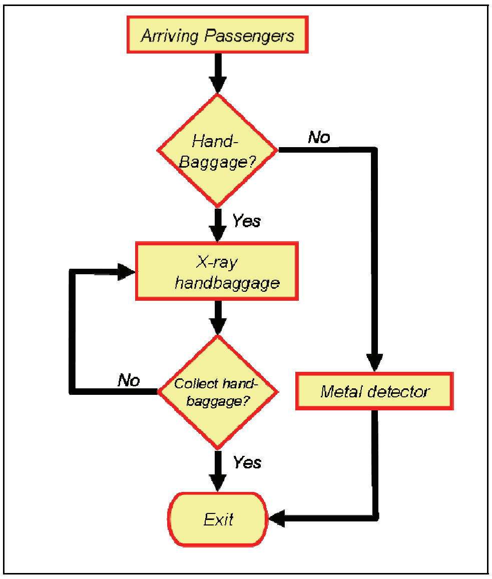


Figure 4: The Security Checks Block Diagram

When passengers enter the Security Check area, they pass through the metal detector while their hand-baggage are checked by X-ray. After taking their hand-baggage, passengers leave the security check area.

3.1.2. The Baggage Area

Figure 5 shows the structure diagram of the Baggage area. This class is implemented in order to reproduce all the operations and activities related to baggage management in the airport terminal (baggage check-in, baggage delivery to/from landing/taking off aircrafts, baggage reclaim, etc.).

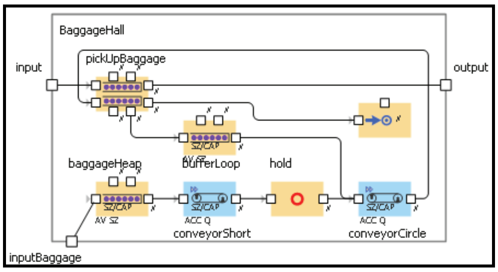


Figure 5: The Baggage Area Structure Diagram

In order to understand more deeply the processes, a block diagram which reproduces in detail the operations related to the baggage hall operations is presented, see Figure 6. Similar approaches have been used for the remaining classes reported in the section 3.1.

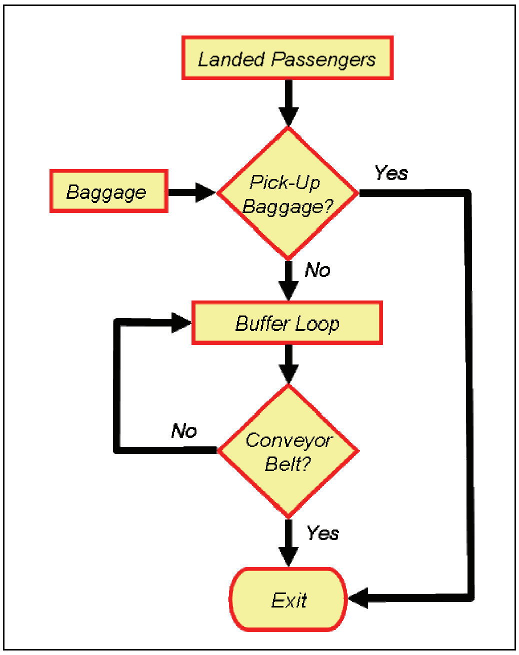


Figure 6: The Baggage Area Block Diagram

3.2. Model verification and validation

The verification of the simulation model implemented is carried out by using an iterative procedure in order to find and eliminate all the possible bugs while the validation is made up by two different steps:

- the first one consists in comparing the historical data about national and international passengers in the period January 2005 – May 2006 and the simulated flow of passengers in the same period;

- the second step compares the real flow of passengers with the flow of passengers obtained by the model for the most important Italian routes.

Figure 7 reports results of the first validation.

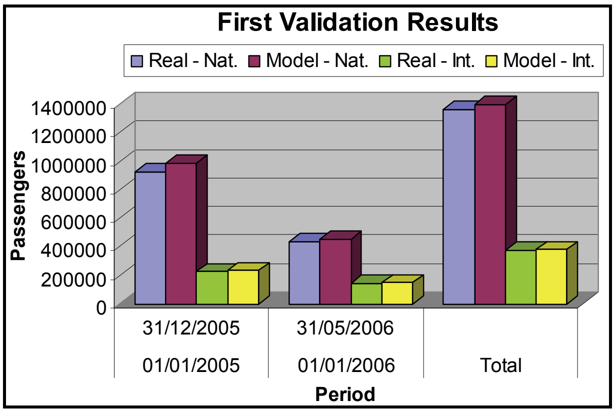


Figure 7: Results of the First Validation

As shown, the difference between real and model results for national passengers is 5.37% while the difference for international passengers is 2.22%: this confirms that the simulation model implemented recreates accurately the real airport terminal.

The simulation run length obtained by using the Mean Sqaure pure Error analysis is 130 days, as reported in Curcio et al. (2007).

4. THE APPLICATION EXAMPLE AND THE DESIGN OF EXPERIMENTS

As before mentioned, the focus of this paper is to implement a model of the Lamezia Terme International airport in order to carry out what-if analysis about its performance under different operative scenarios.

These scenarios are obtained changing system input parameters between specific values. The input parameters (factors) are:

- the number of security control lines (*SCs*) which can assume two different values (2 and 5);
- the number of passport control lines (*PCs*) which can vary from 2 to 5;
- the number of check-in points (*CIs*) which can be changed respectively in 10 and 15.

The combinations of such parameters' values provide different operative scenarios and affect the utilization index (evaluated as the average of the utilization indexes of the of the SCs, PCs and CIs).

For evaluating the effect of each possible parameters combination on the system performance index, the Full Factorial Experimental Design is adopted. Factors and levels adopted for the design of experiments (*DOE*) are reported in Table 1.

Table 1: Factors and Levels of DOE

Factors	***Level 1***	***Level 2***
SC	2 (-1)	5 (+1)
PC	2 (-1)	5 (+1)
CI	10 (-1)	15 (+1)

Each factor has two levels: Level 1 (-1) indicates the lowest value for the factor while Level 2 (+1) represents the greatest value.

In order to test all the possible factors combinations, the total number of the simulation runs is 2^3 (2 levels and 3 factors). Each simulation run has been replicated five times, so the total number of replications is 40 (2^3x5=40).

The output data provided by the simulation model are then studied, according to the various experiments, by means of statistical tools, i.e. the Analysis Of Variance (*ANOVA*), and of several graphical methods.

5. SIMULATION RESULTS ANALYSIS

As before mentioned, the results of the simulation model have been analyzed by means of ANOVA and of several graphical tools.

The ANOVA partitions the total variability of the performance index in different components due to the influence of the factors considered.

According to Montgomery and Runger (2003), the total variability in the data, measured by the total corrected sum of squares SQ_T, can be partitioned into a sum of squares of differences between treatment (factor level) means and the grand mean denoted $SQ_{Treatments}$ and a sum of squares of differences of observations within a treatment from the treatment mean denoted SQ_E, as reported in equation 7.

$$SQ_T = SQ_{Treatments} + SQ_E \tag{1}$$

More in detail, the difference between observed treatment means and the grand mean defines differences between treatments, while observations' differences within a treatment from the treatment mean can be due only to random errors. As a consequence, examiners can understand how each factor impacts on the performance index introducing an analytical relation (called *meta-model* of the simulation model) between the performance index and factors. In particular, the relation for a three-factor-factorial experiment is:

$$\begin{aligned} Y &= \beta_0 + \sum_{i=1}^{3} \beta_i x_i + \sum_{i=1}^{3} \sum_{j>i}^{3} \beta_{ij} x_i x_j + \\ &+ \sum_{i=1}^{3} \sum_{j>i}^{3} \sum_{k>j}^{3} \beta_{ijk} x_i x_j x_k + \varepsilon_{ijkn} \end{aligned} \tag{2}$$

where:

- Y_{ijkn} is the performance index (utilization index)

- β_0 is a constant parameter common to all treatments;
- $\sum_{i=1}^{3} \beta_i x_i$ are the three main effects of factors;
- $\sum_{i=1}^{3} \sum_{j>i}^{3} \beta_{ij} x_i x_j$ are the three two-factor interactions;
- $\sum_{i=1}^{3} \sum_{j>i}^{3} \sum_{k>j}^{3} \beta_{ijk} x_i x_j x_k$ represents the three-factor interaction;
- ε_{ijkn} is the error term;
- n is the number of total observations.

In this research study, ANOVA is adopted for a twofold reason:

- during the first step, it is used as a screening tool in order to determine which factors are most significant on the performance index (*sensitivity analysis*);
- subsequently, ANOVA allows to make analysis about the most significant factors in order to develop the input-output meta-model and to explain interactions between them.

Table 2 reports the simulation results; the first three columns report the experimental design matrix while the last column contains the simulation results in terms of utilization index (indicated as ATCF, five replications). Table 3 reports the sensitivity analysis results provided by Minitab™: the non-negligible effects are characterized by a *p-value* ≤α where *p* is the probability to accept the negative hypothesis (the factor has no impact on the performance index) and α=0.05 is the confidence level adopted in the analysis of variance. In this table:

- the first column reports the sources of variations;
- the second column is the degree of freedom (*DOF*);
- the third column is the Sum of Squares;
- the 4th column is the Mean Squares;
- the 5th column is the Fisher statistic;
- the 6th column is the p-value.

In this case the most significant effects are the main effects and the second order effects because their p-value is lower than the confidence level. More in detail, Figure 8 (the Pareto Chart for the Standardized Effects) shows that the most significant effects are:

- SC;
- CI;
- PC*CI.

Table 2: Experimental design matrix and simulation results

Factors			***Utilization Index, ATCF***				
SC	***PC***	***CI***	***Rep1***	***Rep2***	***Rep3***	***Rep4***	***Rep5***
-1	-1	-1	0.835	0.881	0.857	0.873	0.878
1	-1	-1	0.786	0.763	0.791	0.798	0.765
-1	1	-1	0.836	0.823	0.816	0.850	0.828
1	1	-1	0.769	0.758	0.758	0.745	0.756
-1	-1	1	0.801	0.776	0.784	0.803	0.793
1	-1	1	0.728	0.704	0.720	0.733	0.731
-1	1	1	0.790	0.783	0.771	0.773	0.790
1	1	1	0.693	0.806	0.839	0.689	0.716

Table 3: Sensitivity Analysis Results by Minitab™

Source	***DF***	***AdjSS (10^{-3})***	***AdjMS (10^{-3})***	***F***	***P***
Main Effects	4	65,09	21,69	28,91	0
2-Way interactions	1	6,65	2,21	2,95	0,04
3-Way interactions	1	0,38	0,38	0,51	0,48
Residual Error	25	24,01	0,75		
Total	31				

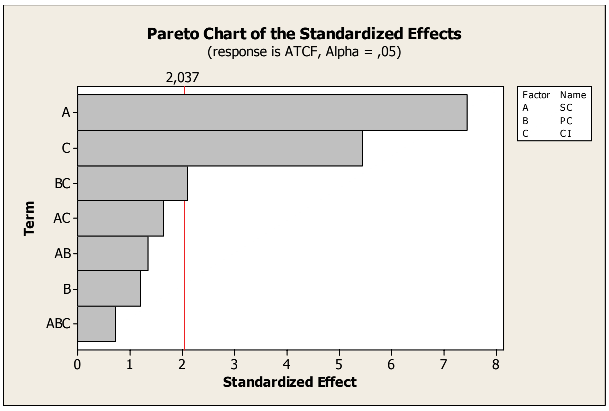


Figure 8: Pareto Chart for the Standardized Effects

This is also confirmed by the Normal Probability Plot of the Standardized Effects reported in Figure 9.

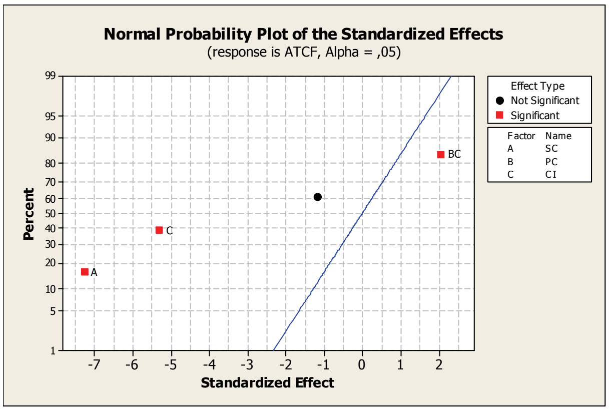


Figure 9: Normal Probability Plot for the Standardized Effect

The input-output meta-model for the airport utilization index is:

$$
\begin{aligned}
ATCF &= 0.7847 - 0{,}03228 * SC + \\
&- 0{,}0052 * PC - 0.02358 * CI + \\
&0{,}00910 * (PC * CI)
\end{aligned}
\tag{3}
$$

Figure 10 shows the utilization index ATCF versus the two main effects: the performance parameter decreases when the number of security check and check-in lines increase because passenger flow is not equally distributed among all the security check and check-in points.

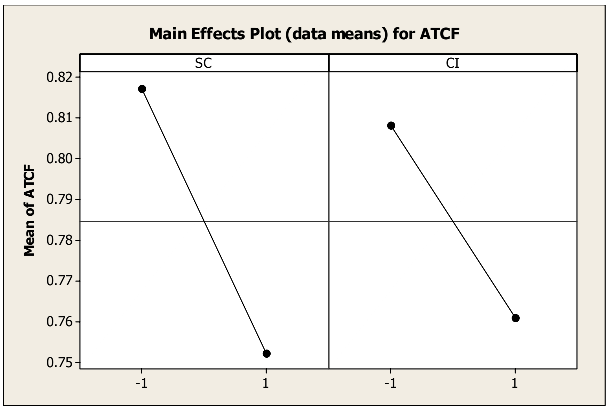


Figure 10: Airport Terminal Capacity Factor versus Main Effects

6. CONCLUSIONS

In this paper a simulation model of a real airport terminal is presented. The software tool adopted for the model implementation is Anylogic™ by *XJ Technologies*. The goal of the research work was the investigation of the system behavior under different operative scenarios. The performance parameter investigated is a mean utilization index including the utilization of check-ins, security control and passports control. The simulation results analyzed by means of ANOVA evidence how input parameters' changes affect the capacity factor of the whole terminal. The final result of the ANOVA is the analytical relation expressing the utilization index as function of the parameters being considered.

REFERENCES

Andreatta, G., Brunetta, L., Righi, L., 1999. An Operations Research Model for the Evaluation of an Airport Terminal: SLAM. *Air Transport Management*, 5, 161–175.

Bocca, E., Longo, F., Mirabelli, G., Viazzo, S., 2005. Developing data fusion systems devoted to security control in port facilities. *Proceedings of the Winter Simulation Conference*. December 03-08, Orlando (Florida, USA).

Brunetta, L., Romanin-Jacur, G., 1999. Passenger and baggage flow in an airport terminal: a flexible simulation model. *Air Traffic Management*, 6, 361–363.

Curcio, D., Longo, F., Mirabelli, G., Papoff, E., 2007. Passengers' Flow Analysis and Security Issues in Airport Terminals Modeling & Simulation applied to Security Systems. *Proceedings of the 21st European Conference on Modeling and Simulation*, pp. 374-379. June 04-06, Prague (Czech Republic).

Curcio, D., Longo, F., Mirabelli, G., 2007. Airport Terminal Management using Web-based Simulation. *Proceedings of the EUROSIW 2007*. June 18-20, Genoa (Italy).

Fayez, M.S., Kaylani, A., Cope, D., Rychlik, N., Mollaghasemi, M., 2008. Managing airport operations using simulation. *Simulation*, 2, 41–52.

Gatersleben, M.R., Van der Weij, S.W., 1999. Analysis and simulation of passengers flows in airport terminal. *Proceedings of the 1999 Winter Simulation Conference*. December 05-08, Phoenix (Arizona, USA).

Glasser, U., Rastkar, S., Vajihollahi, M., 2006. Modeling and Analysis of Aviation Security Procedures. ed. *Creative Commons Attribution-NonCommercial-NoDerivs License*. CA: 1–28.

Jim, H.K., Chang, Z.Y., 1998. An airport passengers terminal simulator: a design tool. *Simulation Practice and Theory*, 6, 387–396.

Klein, U., 2000. Simulation-based distributed systems: serving multiple purposes through composition of components. *Safety Science*, 35, 29–39.

Rossiter, A., Dresner, M., 2004. The impact of the September 11th security fee and passenger wait time on traffic diversion and highway fatalities. *Air Transport Management*, 10, 227–232.

Sherali, H.D., Hobeika, A.G., Trani, A., Byung, J.K., 1992. An Integrated Simulation and Dynamic Programming Approach for determining Optimal runway exit locations. *Management Science*, 38 (7), 1049–1062.

Snowdon, J.L., MacNair, E., Montevecchi, M., Callery, C.A., El-Taji, S., Miller, S., 2000. IBM Journey Management Library: An Arena System for Airport Simulations. *Operational Research*, 51 (4), 449–456.

Tosic, V., 1992. A Review of Airport Passenger Terminal Operations Analysis and Modeling. *Transportation Research*, 26A (1), 3–26.

Verbraeck, A., Valentin, E., 2002. Simulation Building Blocks for Airport Terminal Modeling. *Proceedings of the 2002 Winter Simulation Conference*, pp. 1199-1206. December 08-11, San Diego (California, USA).

Yfantis, E.A., 1997. An intelligent baggage-tracking system for airport security. *Engineering Applications of Artificial Intelligence*, 10 (6), 603–606.

AUTHORS BIOGRAPHIES

ANTONIO CIMINO was born in Catanzaro (Italy) in October the 1th, 1983. He took his degree in Management Engineering, summa cum Laude, in September 2007 from the University of Calabria. He is currently PhD student at the Mechanical Department of

University of Calabria. His research activities concern the integration of ergonomic standards, work measurement techniques, artificial intelligence techniques and Modeling & Simulation tools for the effective workplace design. He collaborates with the Industrial Engineering Section of the University of Calabria to research projects for supporting innovation technology in SMEs.

DUILIO CURCIO was born in Vibo Valentia (Italy), on December the 15th, 1981. He took the degree in Mechanical Engineering from University of Calabria (2006). He is currently PhD student at the Mechanical Department of University of Calabria. His research activities include Modeling & Simulation and Inventory Management theory for production systems and Supply Chain design and management. He collaborates with the Industrial Engineering Section of the University of Calabria to research projects for supporting Research and Development in SMEs.

GIOVANNI MIRABELLI was born in Rende (Italy), on January the 24th, 1963. He took the degree in Industrial Technology Engineering from University of Calabria. He is currently researcher at the Mechanical Department (Industrial Engineering Section) of University of Calabria. His research interests regard work measurement and human reliability.

ENRICO PAPOFF was born in Naples (Italy) on February the 03rd, 1948. He took the degree in Mechanical Engineering from University of Napoli Federico II, in 1973. He is currently Associate Professor at the Mechanical Department (Industrial Engineering Section) of the University of Calabria. His research interests regard project management and business plans.

ON THE USE OF OPTICAL FLOW TO TEST CROWD SIMULATIONS

David J. Kaup(a), **Thomas L. Clarke**(b), **Linda Malone**(c), **Rex Oleson**(d), **Mario Rosa**(e)

(a,b) Institute for Simulation and Training, 3100 Technology Parkway, Orlando, FL 32826
(a,b) Mathematics Department, University of Central Florida, Orlando, FL 32816-1364
(c) Department of Industrial Engineering and Management, University of Central Florida, Orlando, FL 32816-2450
(d,e) Institute for Simulation and Training, 3100 Technology Parkway, Orlando, FL 32826

(a)kaup@mail.ucf.edu, (b)tclarke@ist.ucf.edu, (c)lmalone@mail.ucf.edu, (d)quazerin@juno.com, (e)mer4m@yahoo.com

ABSTRACT

A method for extracting estimates of crowd movement from videos of the crowd was reported at EMSS 2007. The goal of this method is to provide a means to validate the accuracy of crowd models such as the Helbing-Molnar-Farkas-Vicsek (HMFV) model and to provide a means to refine model parameters. The method uses optical flow extracted from the video with an empirical calibration constant to convert the optical flow to boundary crossing rates of crowd movement. A simple proportional relationship between optical flow and crossing rates of people was postulated on the basis of theoretical considerations. This paper reports on further crowd observations designed to confirm the simple form of the proportionality constant. While the research is on-going, preliminary results support the relationship.

Keywords: optical flow, people flux, culture differences, crowds, social force model.

1. BACKGROUND

1.1. Crowd Modeling

In our research (Kaup, Fauth, Walters, Malone, and Clarke 2006) we have been focusing on a continuous space crowd model, which we refer to as the Helbing-Molnar-Farkas-Vicsek (HMFV) model. This model is described in Helbing, Farkas, and Vicsek (2000). In the HMFV model, each pedestrian feels, and exerts on others, two kinds of forces, “social" and physical. The social forces do not have a physical source; rather, they reflect the intentions of a pedestrian not to collide with other people in the room or with walls and also to move in a specific direction (e.g., towards an exit) at a given speed. Symbolically, the force exerted on pedestrian *i* by pedestrian *j* has the form:

$$\vec{f}_{ij} = \vec{f}_{social\ repulsion} + \vec{f}_{pushing} + \vec{f}_{friction} \qquad (1)$$

The first term in Equation (1) describes the social force,

$$\vec{f}_{social\ repulsion} = \mathit{Constant}_{\boldsymbol{B}} \times \mathit{exponential(interpersonal\ distance)}_{radial\ direction} \qquad (2)$$

while the second and third terms describe the physical forces of pushing and sliding friction between the two pedestrian bodies.

$$\vec{f}_{pushing} = \mathit{Constant}_{\boldsymbol{k}} \times \mathit{threshold(interpersonal\ distance)}_{radial\ direction} \qquad (3)$$

$$\vec{f}_{friction} = \mathit{Constant}_{\boldsymbol{\kappa}} \times \mathit{threshold(interpersonal\ distance)}_{tangential\ direction} \qquad (4)$$

The form of the latter two terms ensures that they vanish when the pedestrian bodies are not in physical contact. An expression similar to equations (3) and (4) holds for a force between a pedestrian and a wall or another immobile obstacle (e.g., a column) in the room.

1.2. Comparison to Real Crowds

Video imagery of moving crowds was obtained for the purpose of providing experimental verification of predictions of crowd models such as the Helbing-Molnar-Farkas-Vicsek (HMFV) model. With Institutional Review Board (IRB) approval, cameras were set up at university and other public events. The field of view of the cameras was aimed at exit points or other locations that could be easily simulated.

The optical flow field can be extracted from a video using the Lucas-Kanade algorithm (Lucas and Kanade 1981) as implemented in the Intel OpenCV (Intel 2001) image processing library. A frame from such a video taken at a local church is shown in Figure 1. A 400-frame sequence (13.3 seconds) from this video was used to create the optical flow field shown in Figure 2.

We make the following assumptions concerning the relation between apparent optical flow and actual crowd motion or flux:

1. Any motion is due only to that of people in the crowd
2. The mapping between physical space in which people move and the image space is assumed to be a simple scaling
3. The reflectivity and illumination of people and other elements in the scene are taken to be uniform

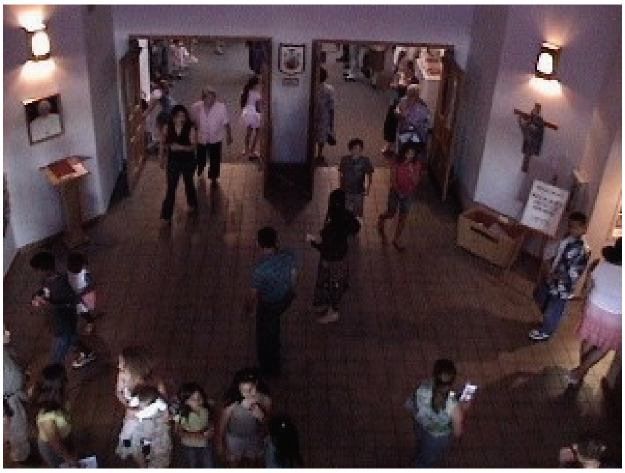


Figure 1: Hispanic Crowd Exiting Church at Frame 200.

It can be shown under these assumptions that the rate at which people cross a boundary is proportional to the optical flow in a region surrounding that boundary. Figure 3 illustrates this; the number of people F_y crossing a horizontal boundary in a vertical direction is simply proportional to the average vertical optical flow V_y in the shaded region surrounding the boundary. A similar relation holds for F_x and V_x in the horizontal direction.

The proportionality constant between $F_{x|y}$ and $V_{x|y}$ is not dependent on the number of people in the crowd and is primarily a function of the camera distance.

To determine the proportionality constant between optical flow and actual crowd motion, hand counts of people crossing the boundaries of a 3 by 4 grid were pre- formed on selected segments of the videos taken at an American football game. These counts were compared with optical flow calculated on a 6 by 8 grid. Use of a doubly fine grid for the optical flow facilitated the analysis of the relation shown in Figure 3. Limited comparisons confirmed the relation between flux of people and optical flow (Clarke, Kaup, Malone, Oleson, and Rosa 2007).

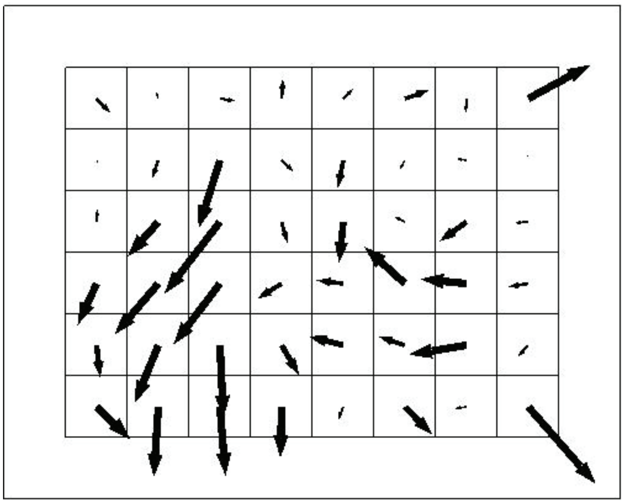


Figure 2: Optical Flow Field for Hispanic Case from Frames 1-400.

2. CURRENT RESULTS

To further confirm the relation between flux of people in a crowd and optical flow, further videos have been acquired and analyzed for optical flow and subjected to hand counting of people flux.

These videos came from the following venues:

1. A primarily Hispanic congregation exiting a church
2. A primarily Anglo congregation existing a church

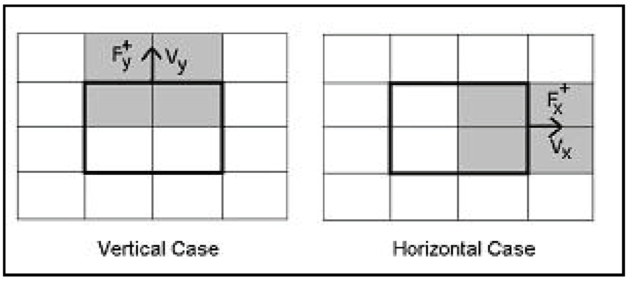


Figure 3: The Geometrical Relation between Optical Flow and Boundary Crossing Rate.

2.1. Hispanic Case

The video taken of the Hispanic crowd exiting the church was 1100 frames long. The first 400 frames (13.3 seconds) and the last 400 frames were separated out for analysis. Figure 1 shows the 200th frame in the center of the first analysis interval. The optical flow field calculated from frames 1-400 for the Hispanic case is shown in Figure 2.

The 3x4 grid boundary crossing rate was hand-counted from frames 1-400 and plotted versus the optical flow. The relation was essentially random for this case. This is not too surprising, however. Examination of frames 1-400 shows the crowd is milling about during this time interval of 13.3 seconds as suggested by Figure 1 and the optical flow field in Figure 2. Whence the time interval used was too long to effectively pick out any motion other than the average; in this case the average is essentially zero.

Figure 4 shows the single frame 900 of the Hispanic-exit case, and Figure 5 shows the optical flow calculated from frame 701-1100.

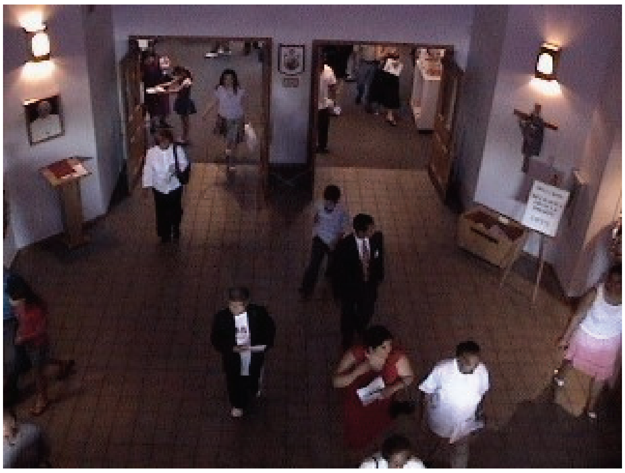


Figure 4: Hispanic Crowd Exiting Church at Frame 900.

The results of plotting 3 by 4 grid crossing rates versus optical flow were much better for this case as illustrated in Figure 6. The slope of the best fit line is 0.0092 and the correlation is 0.329. The reason for this relatively good correlation seems to be the purposeful movement of the crowd as shown in Figure 4. While the crowd appears sparse in frame 900 (Figure 4), in the video prior to frame 900 the crowd is much denser.

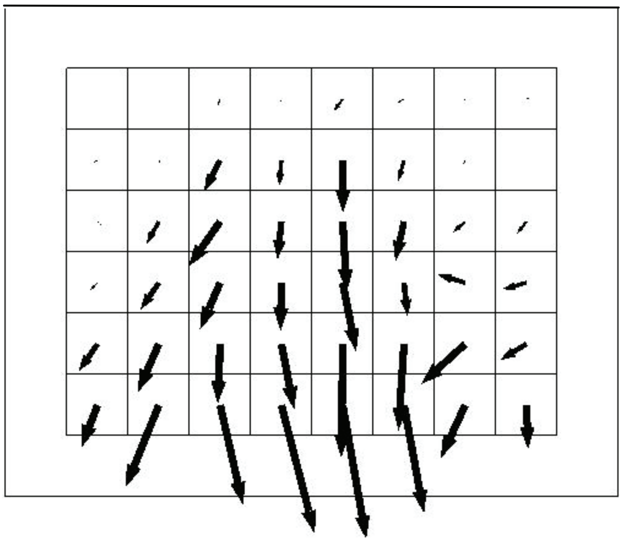


Figure 5: Optical Flow Field for Hispanic Case From Frames 701-1100.

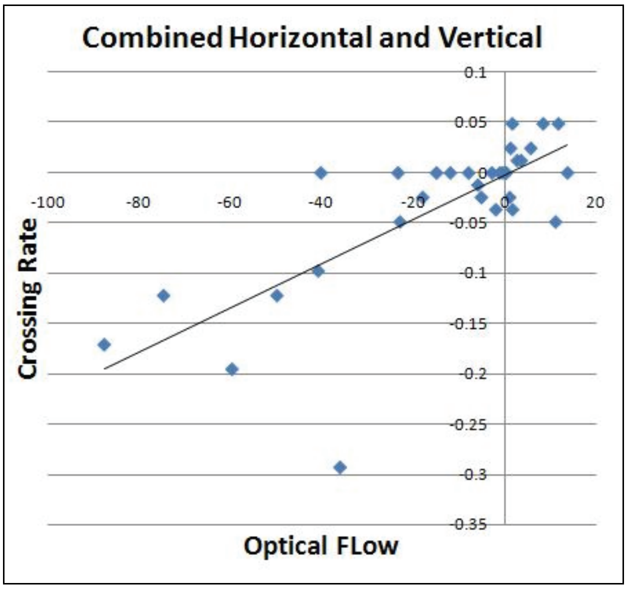


Figure 6: Correlation Plot of 3 By 4 Grid Crossing Rate Versus Optical Flow for Hispanic Frames 701-1100.

2.2. Anglo Case

Figure 7 shows frame 200 of the Anglo (English speaking) congregation leaving the church. The Anglo-case video was 5400 frames long (180 seconds). The camera angle and distance were also different from the Hispanic case.

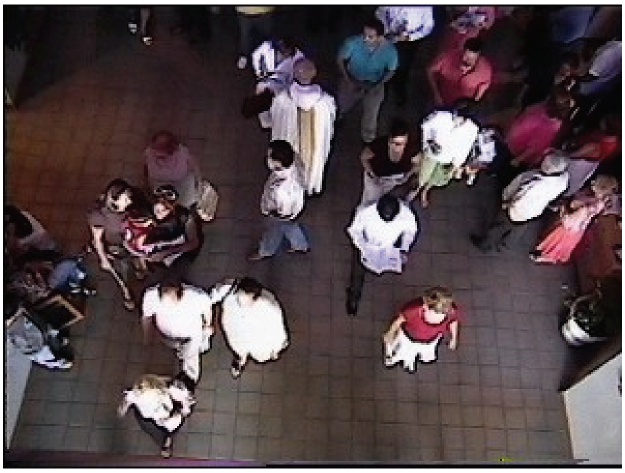


Figure 7: Anglo Crowd Exiting Church at Frame 200.

Figure 8 shows the optical flow field for the Anglo case for frames 1-400. Note the minimum in the velocity at the position corresponding to a stationary priest who can be seen in Figure 7.

Despite the presence of the priestly obstacle in the Anglo case, the correlation between crossing rate and optical flow was good as can be seen in Figure 9.

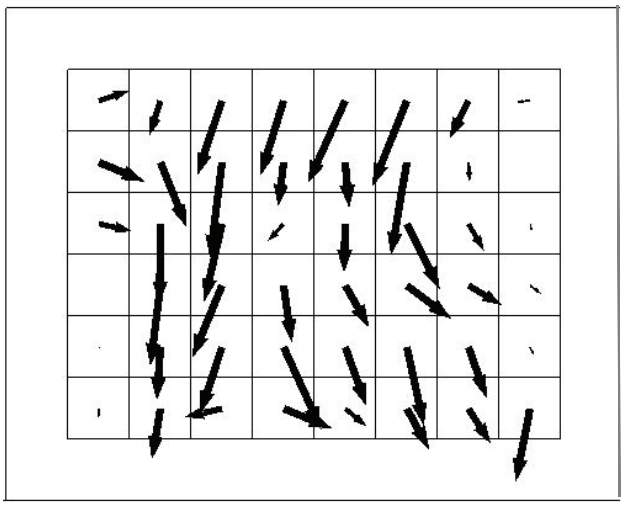


Figure 8: Optical Flow Field for Anglo Case from Frames 1-400.

The slope of the linear fit in Figure 9 is 0.0036 and the correlation coefficient is 0.467. The relative magnitudes of the slopes in Figures 6 and 9 are consistent with the difference in camera distance and angle for the two cases.

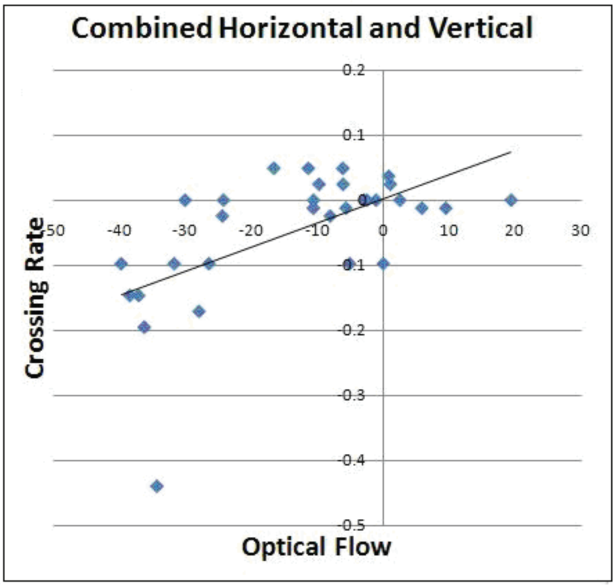


Figure 9: Correlation Plot of 3 By 4 Grid Crossing Rate Versus Optical Flow for English Frames 1-400.

Figures 10 and 11 show frame 5200 (173.3 seconds into the video) and the optical flow field for frames 5001-5400. The crowd had become fairly sparse and was milling about more or less randomly by this time, so the crossing rate/optical flow relation was random much like the first Hispanic case presented.

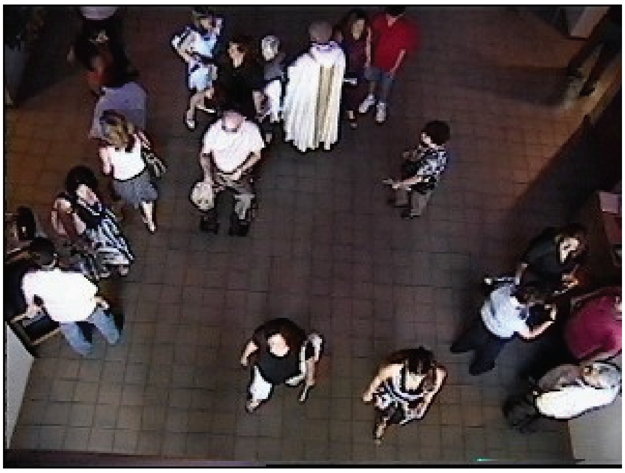


Figure 10: Anglo Crowd Exiting Church at Frame 5200.

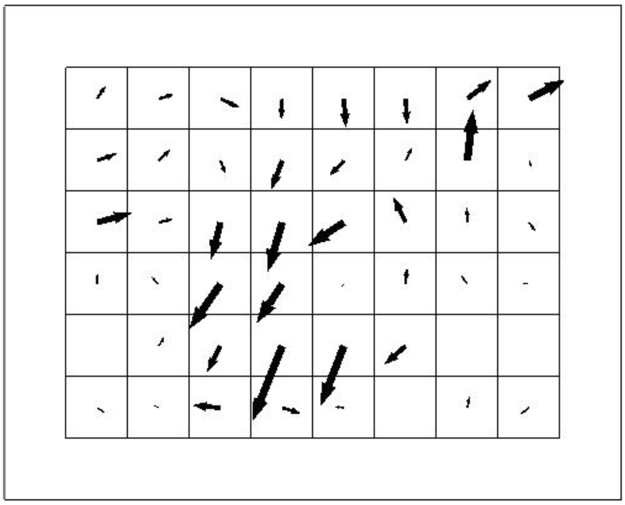


Figure 11: Optical Flow Field for Anglo Case from Frames 5001-5400.

3. CONCLUSION

On the basis of preliminary results, the correlation between hand counted boundary crossing rates and optical flow is high and statistically significant when the crowd is moving in a definite direction. When the crowd is milling about the correlation is generally poor. Thus, the easily measured optical flow could stand in place of time consuming manual counts of the flux of people moving in crowds to provide experimental validation of crowd models. The dependence of the proportionality constant on viewing distance has the form expected. It is also clear that one should investigate comparisons at smaller time intervals (perhaps even as small as 10 frames) wherein one possibly could detect fluxes associated with random milling motion.

ACKNOWLEDGMENTS
We thank P. Kincaid, B. Goldiez and R. Shumaker of the Institute for Simulation and Training for their interest and discussions throughout different stages of this work. This research has been supported in part by the National Science Foundation under Grant No. BCS-0527545 and by the U.S. Army Research Development and Engineering Command, Simulation Technology Center, Contract N61339-02-C-0107.

REFERENCES

Clarke, T.L., Kaup, D.J., Malone, L., Oleson, R. & Rosa, M., 2007. Crowd model verification using video data. *Proceedings of EMSS 2007*, paper no. 31. October 4-6, Bergeggi (Italy).

Helbing, D., Farkas, I. & Vicsek, T., 2001. Simulating dynamical features of escape panic. *Nature*, 407, 487-490.

Intel, 2001. *Open Source Computer Vision Library: Reference Manual.* Intel. Available from: www.itee.uq.edu.au/~iris/CVsource/OpenCVreferencemanual.pdf [accessed 15 February 2006].

Kaup, D.J., Fauth, J.E., Walters, L.J., Malone, L. & Clarke, T., 2006. Simulations as a mathematical tool. *SIAM Conference on the Life Sciences,* July 31-August 4, Raleigh (North Carolina, USA).

Lucas, B.D. & Kanade, T., 1981. An interactive image registration technique with an application to stereo vision. *Proceedings of Image Understanding Workshop*, pp. 121-130. April 23, Washington, DC (USA).

AUTHORS BIOGRAPHY

DAVID J. KAUP is Provost Distinguished Research Professor in mathematics at the University of Central Florida with a joint appointment at the Institute for Simulation and Training. He has a PhD in Physics from the University of Maryland. He was a professor at Clarkson University before coming to UCF. His research interests are in the area of non-linear waves, in particular solitons, and he has been recently applying this background to the area of modeling the dynamics of crowds.

THOMAS L. CLARKE is Principal Mathematician at the Institute for Simulation and Training of the University of Central Florida and is also Associate Professor in the Modeling and Simulation program. He has a PhD in Applied Mathematics from the University of Miami and worked at NOAA before coming to UCF. He has a diverse background in applying mathematics to simulation and has investigated areas such as the application of catastrophe theory and nonlinear dynamics.

LINDA C. MALONE is a Professor in the Industrial Engineering Department at University of Central Florida. She got her B.S. and M.S. degrees in mathematics (at Emory and Henry College and University of Tennessee respectively) and her Ph.D. degree in statistics from Virginia Tech in 1975. She is the coauthor of a statistics text, has authored or coauthored 74 refereed papers. She was an associate editor of the Journal of Statistical Computation and Simulation for over 25 years and was a founding co-editor of the STATS Magazine. She has served in various offices in statistical organizations including service on the Board of Directors of the American Statistical Association. She was awarded the honor of Fellow of the American Statistical Association.

REX OLESON graduated from Susquehanna University with a B.S. in physics and mathematics. Following four years of software development, he obtained an M.S. in Mathematics from UCF. He is currently working on a Ph.D. in Modeling and Simulation. He is working through IST.

PROXEL-BASED SIMULATION OF QUEUING SYSTEMS WITH ATTRIBUTED CUSTOMERS

Claudia Krull[(a)], Wenjing Xu[(b)], Graham Horton[(c)]

[(a)(b)(c)]University of Magdeburg, Department of Simulation and Graphics

[(a)]claudia@sim-md.de, [(b)]wenjing_de@yahoo.com, [(c)]graham@sim-md.de

ABSTRACT

This paper describes a state space-based simulation method for queuing systems with attributed customers. The approach extends a previous version, which was designed for only one customer class, enabling the simulation of a larger group of queuing models. The work is motivated by the need for exact solutions for queuing systems where no analytical solution is available. In this paper, the original Proxel-based queuing simulation method is extended to incorporate attributed customers, concentrating on efficient coding and storage strategies to dampen the state space explosion. The attribute classes priority, deadline and processing time are implemented. Experimental results indicate the maximum number of attribute values that is still feasible. Some interesting statistics are presented, which would be hard to obtain using traditional simulation methods. The presented method can yield deterministic results for a larger number of queuing systems that cannot be easily solved analytically.

Keywords: state space-based simulation, multiclass queuing systems, attributed customers.

1. INTRODUCTION

Queues are a part of our everyday life. Besides the obvious queues of human customers, they are widely used in telecommunications and computer systems. Therefore, queuing analysis is a well-researched area of mathematical modeling. The usual approach to a queuing problem is to devise an appropriate formal representation, find a match among the already known classes of queuing systems or create a new one and derive measures for the queuing system's performance analytically.

For a number of classes of queuing systems, no analytical expression for performance measures like throughput or server utilization is available. In such cases, simulation is the preferred way to tackle the problem. However, discrete event-based simulation may need to perform many replications to obtain estimates for the results of one single specified queuing system.

The recently developed method of Proxel-based simulation (Horton 2002; Lazarova-Molnar 2005) can feasibly conduct state-space based analysis of discrete stochastic systems and yield accurate deterministic results with reasonable effort. The Proxel results can compete with typical analytical ones, and the method is not inherently limited in the type of system.

Proxels have already been successfully applied to the simulation of queuing systems with one type of customer (Krull and Horton 2007). However, many queuing problems implement attributed customers to enable more advanced queuing strategies than FIFO.

The object of this paper is to introduce attributes into a Proxel-based queuing system simulation, test the resulting performance and show the suitability of the approach. If successful, more classes of queuing problems can be tackled using Proxel-based simulation. We believe that this could provide useful help for queuing analysts. Proxel-based simulation can provide deterministic solutions for a specific queuing systems performance measures, when no analytical solution is available. One can obtain rough estimates at low cost or results of arbitrary accuracy at higher computation cost. Furthermore, the method yields a transient solution of the system, which can for example help detect the possibility of infinite postponement when implementing SJF (*shortest job first*) as a queuing strategy.

2. STATE OF THE ART

2.1. Queuing System Analysis

Classical queuing analysis takes a real queuing system and derives a queuing model by identifying the arrival process, the service process and other system specifications. The performance measures for the queuing model are then calculated using known formulae for the given class of queuing system. The performance measures of a queuing system are usually expressed as scalar measures such as the system throughput. (Bolch et. al 2006, Gross and Harris 1998)

Queuing systems that contain generally distributed processes are often hard to handle analytically. For most of these classes no general solution exists. Discrete event-based simulation can be used to obtain stochastic estimates for these systems performance measures. However, unless a lot of computation cost is invested, the accuracy of these results is not comparably to analytical solutions.

An important application area of queuing analysis is the modeling of data traffic in networks, for example in the internet. The models consist of multiple queuing systems that are connected to form queuing networks. As stated in Cremonesi, Schweitzer and Serazzi (2002) the traffic consists of customers with very different resource requirements (attributes), resulting in so-called multiclass queuing networks. The paper introduces a modeling framework for approximate solutions, because it is not feasible to derive exact solutions for problems with large state spaces. Exact solutions to multiclass queuing networks are only possible for special classes (Casale 2006). These two examples show the importance of attributed customers, but also their impact on the complexity of exact solution methods.

One approach towards the analysis of queuing networks is the decomposition into single queuing systems and combination of their solutions. In Heindl (2001) one such approach is described for a special class of queuing systems. A decomposition approach for multiclass queuing networks is described in Whitt (1994). The restriction of the decomposition approaches lies in the goal to describe the performance measure results using mathematical analysis. This makes special solutions necessary for the many classes of queuing systems. However, to obtain exact solutions decomposition is the only feasible way to reduce the models' state space to controllable size. Therefore, a generally applicable and exact solution method for single queuing systems is still of interest.

2.2. Proxel-based Queuing System Simulation

Discrete event-based simulation is one way to derive estimates for the scalar performance measures, if no analytical solution exists. However, the results are of stochastic nature and only applicable to the single specified queuing system investigated. The simulation itself can get very expensive for stiff queuing models.

In (Hasslinger and Kempken 2006) an approach is presented for the transient analysis of a queuing system. Transition equations for discrete time points are derived and only the relevant state transitions at arrival and service instances are considered. The approach seems promising, and according to the authors can be extended to a number of queuing problems. However, the example in the paper is small, considering the two discretization points for the arrival and service distributions. The transition equations and the system state space can get complex when the problem gets larger, since all possible combinations of the arrival and service of a customer have to be considered.

The recently proposed Proxel-based simulation of queuing systems has several advantages. It can yield deterministic results for the performance measures of any queuing system. The accuracy of the results can be controlled. In addition to the steady state performance measures, the Proxel method also yields a transient solution, containing the probability of every possible system state at every investigated simulation time step. This cannot be easily obtained using common simulation techniques.

Proxel-based simulation is a state space-based simulation method that scans the possible system development paths in discrete time steps. (Horton 2002, Lazarova-Molnar 2005) In contrast to DES it follows all possible development paths, tracking their respective probabilities and building up a discrete-time Markov chain. This method ensures the discovery of rare development paths. The disadvantage of Proxels is, that through expanding the discrete system states by supplementary age variables, the resulting Markov chain can be of immense size. This so-called state-space explosion limits the applicability of Proxels to models with few discrete states, which is usually the case for queuing models.

The current implementation of the Proxel-based queuing system simulator is restricted to customers without attributes. This limits the possible queuing strategies to FIFO and other simple queuing strategies. However, many real systems use more sophisticated strategies for ordering their customers. To enable the Proxel-based simulation of more realistic queuing models, attributed customers are essential.

The basic idea of the extension to attributed customers and one example attribute implementation have been described in Wenjing Xu (2008).

3. ADDING ATTRIBUTED CUSTOMERS

This section describes the steps which are necessary to include attributed customers in the Proxel-based simulation of queuing systems.

We selected three static attributes without preemption policy for implementation. Adding variable attributes or preemption would further enlarge the state space. We also decided to limit the approach to ordering strategies with only one attribute, again to dampen state space explosion.

3.1. Attribute Choice

Common queuing strategies often sort the arriving customers by intrinsic priorities or some given time restrictions. Therefore, the following three attributes were chosen for the implementation and test of the method: *priority*, *processing time* and *deadline*.

A customer's priority is a measure of urgency or importance. It is given by an integer, where a smaller number implies a higher priority.

A customer's processing time holds the information how long the servicing of the customer will take. The processing time is usually given by a real value, but it can be converted into an integer by discretization using the Proxel simulation step Δt. This reduces memory cost.

The attribute deadline is a measure of urgency, which is again given by a real value. The deadline can also be expressed as the difference between the current simulation time and the customer's deadline. Doing this limits the attribute's value set and thereby reduces the system state space. The 'time to deadline' can also be converted to an integer by discretizing this distance using the discrete simulation time step Δt.

3.2. Coding and Storage Strategies

All three attributes can be expressed as integer values. The processing time and time to deadline can have possibly large value sets depending on the discretization time step Δt and the maximum support of the distribution.

By adding attributes to the customers in the queuing system, the discrete state space of the model will be increased significantly. This has two effects. First of all, the memory requirement of the algorithm will grow, because the number of Proxels is increased with a greater state space and because the size of a Proxel grows when one replaces a scalar queue length by an integer array holding the queued customers attributes. Secondly, the computational bottleneck of the Proxel-based method is the retrieval of already created Proxels. A newly created Proxel has to be compared to a possibly large number of existing Proxels in the storage container, requiring a comparison of all elements of these Proxels including discrete state space and the age vector. Therefore, a clever storage of the Proxels and the connected discrete system states is necessary to counter these effects.

The attribute of a customer in service can be stored in an array the size of the number of servers. The attributes of the customers in the queue are also stored in an integer array with a maximum length given by the program, that will most likely not be reached. After testing several options, this has shown to be the most efficient storage strategy for saving memory and retrieval time. The queue array contains the attributes of the customers in the order that they have in the queue.

The customer priority can be coded more efficiently than by the enumeration of all customers priorities in the queue. By listing the number of customers of each priority class in the array instead, the size of the array can be limited to the number of priority classes, which is most likely less than the maximum queue length. The example queue ‘111222233555’ could be expressed by ‘34203’. This saves memory and also shortens the time needed to compare two Proxels in the retrieval bottleneck.

Processing time and time to deadline can be enumerated for all queued customers, since there will not be as many customers with the same values.

3.3. Adapting Algorithm

The Proxel-based queuing system simulation algorithm also needs to be adapted slightly.

The attributes of the customers in the queuing system are fixed upon their arrival. Therefore, the formerly single arrival Proxel will now be split into as many Proxels as possible values of the chosen attribute (e.g. number of priority classes, or each possible discretized service time). In each of these Proxels, the queue has to be reordered, meaning the attribute of the arriving customer has to inserted into the queue array at the appropriate position.

The time to deadline and priority attribute have no influence on the processing of the customer. Fixing the processing time of a customer upon arrival turns the actual service into a deterministic process. This reduces the number of possible service time development paths to one, reducing the overall system state space. This decrease might compensate some of the increase caused by splitting each arrival Proxel.

3.4. Adapting Interface

The interface of the Proxel-based queuing simulation tool has to be changed to enable the selection and specification of customer attributes. In the adapted interface the user can choose to include customer attributes in the queuing system. A separate dialog enables a selection of the attributes. The number of priority levels and a probability distribution for the time to deadline can also be specified.

The output part of the interface was not adapted to include statistics of attributed customers. It only holds the overall system performance measures. Due to complexity, the measures calculated with regard to the attributes (see next subsection) are output into files.

3.5. Calculating Performance Measures

The inclusion of customer attributes into the queuing system specification enables the calculation of more sophisticated performance measures. Examples of newly possible measures are the following:

- Priority - average and extreme waiting times per priority class
- Deadline - the number of late customers and cumulative lateness
- Processing time - minimum and maximum waiting time for each possible processing time

All of these performance measures can be calculated on the fly or using the steady state result of the Proxel algorithm. Their calculation does not influence the algorithms performance significantly.

4. EXPERIMENTAL RESULTS

This section describes some experiments conducted to show that the inclusion of attributes into Proxel-based queuing system simulation is possible, and to test the effect of adding attributes on the algorithms performance. Furthermore we want to show some interesting results that can be obtained only by including attributes in the Proxel-based simulation.

The first experiment simulates a queuing system using the SJF (shortest job first) scheduling strategy. We obtain a distribution of the waiting time over the possible processing time values. This result is not easily computable using existing queuing analysis and simulation methods. The investigated queuing system has a Markovian arrival process with a rate of one customer per minute and a normally distributed service process with the parameters μ=0.5 and σ=0.1. Figure 1 shows the average waiting time for the different discretized values of the service time distribution. The waiting time of customers with short processing times is small, as expected and the waiting time of customers with larger processing times is longer. The steeper increase around the expected value of the service time is due to the higher number of customers created. These customers with equal discrete processing time delay each other when they are ordered according to FIFO.

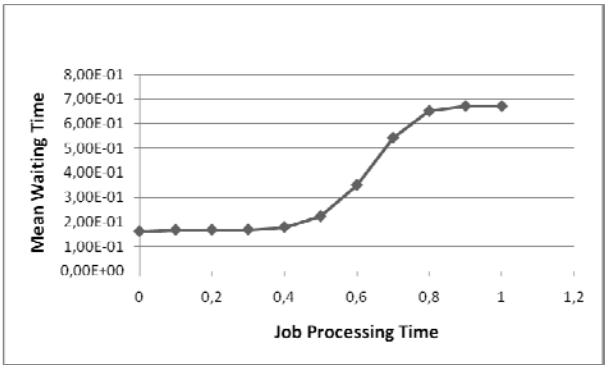


Figure 1: Waiting Time Distribution over Processing Time

The second experiment investigates the effect of adding different processing time distributions on the performance of the Proxel-based simulation method. The arrival process is set to be Markovian with a rate of one customer per minute. The service process is normally distributed with a mean of 0.5 and the standard deviation varied from 0.01 to 0.08. This results in 2 to 12 discretization time steps of the distributions as possible attribute values. Proxel simulations are performed using the same processing time distribution with and without attributes. Figure 2 shows the development of the runtime for the different service time distributions. The runtime of the algorithm without attributed tokens (grey line) is hardly affected by the change in service time distribution. The runtime of the simulation incorporating attributed customers (black line) rises from two seconds to several hundred. The increasing number of discretization time steps of the service time distribution increases the attributes value space and thereby increases the system state space.

This experiment shows, that adding the attribute service time to the Proxel simulation increases the complexity drastically. The combinatorial effect of enumerating every possible combination of customers waiting in the queue blows up the state space by several orders of magnitude. The effect is more severe, when more different attribute values are possible.

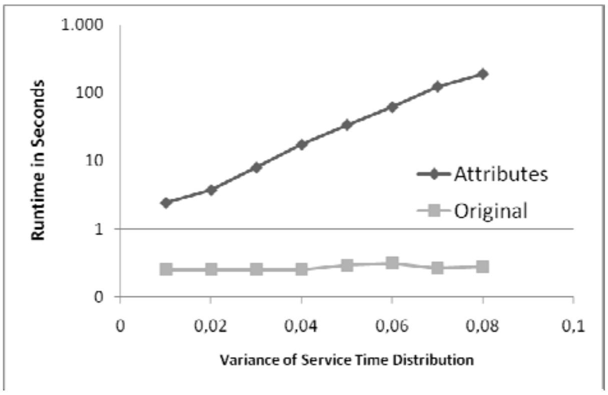


Figure 2: Runtime over Service Time Distribution with and without Attribute

The third experiment investigates the effect on computation cost when adding the attribute priority and increasing the number of priority classes. The tested queuing system has again a Markovian arrival process with an arrival rate of one customer per minute and one server. The service time distribution was chosen to have a rate of two customers per minute. Two different distributions were tested for the service time: Exp(λ=2) and N(μ=0.5, σ=0.1). We want to investigate the effect of the attributes on different queuing systems and compare the storage strategies enumeration of priority classes and count of elements per class. The resulting number of concurrent Proxel elements (as a measure of memory requirement) for both experiments were the same. The overall storage space needed could be reduced to about one half by using the counting strategy, since the storage space needed by one Proxel was reduced.

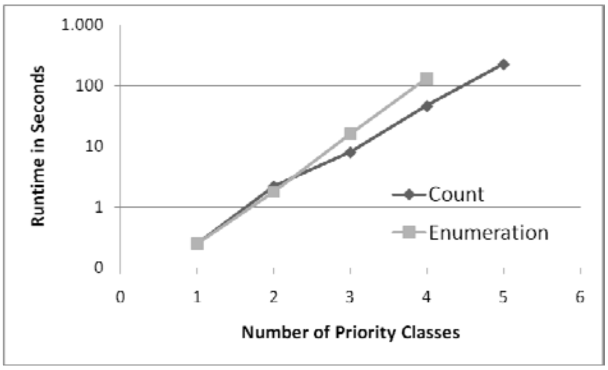


Figure 3: Runtime for Exponential Service Time Distribution and Different Numbers of Priority Classes

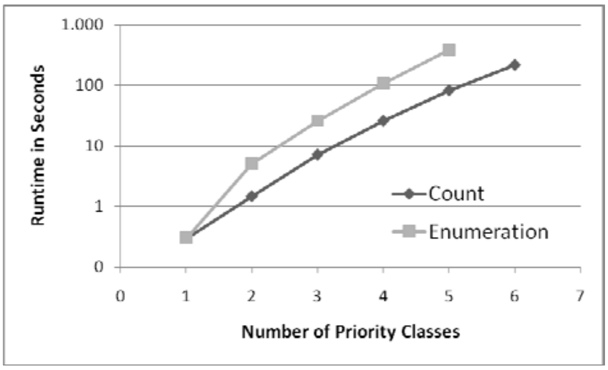


Figure 4: Runtime for Normal Service Time Distribution and Different Numbers of Priority Classes

The development of the algorithm runtime for the M/M/1 queuing system with a growing number of priority classes can be seen in Figure 3. The grey line shows the experiment using enumeration of all queue elements and the black line the development for only counting the elements per priority class. The drastic effect of this small change in the storage scheme is surprising. This shows that the handling of the Proxels and especially the comparison of two individual probability elements is one major bottleneck of the algorithm. When the attributes of all queue elements are stored, these also need to be compared individually when two Proxels are compared. This overhead can be reduced when only the number of elements per priority class need to be compared. The same effect is also visible in Figure 4, where the runtime for the normal service time distribution is depicted.

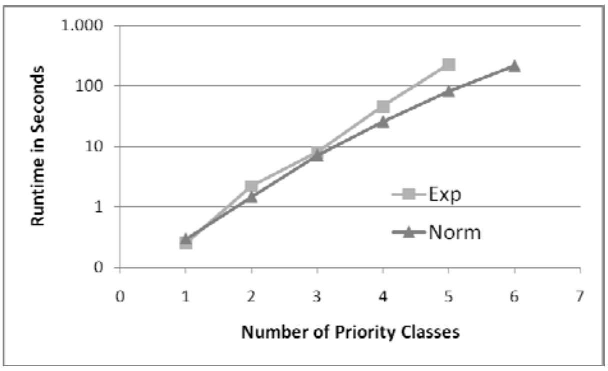


Figure 5: Algorithm Runtime for for Different Servicec Time Distributions with Priority Attribute

In Figure 5 one can see that using a normally distributed service times, the runtime is faster than when using exponential service times. This behavior is quite opposite to that of the original Proxel algorithm, where exponential distributions do not expand the state space due to their constant rate function. The higher runtime costs for the exponential service time distribution are caused by a larger state space, due to longer possible queue length when having a service time distribution with a greater variance.

This experiment shows, that the increase in state space by adding age variables for a non-Markovian distribution is less severe than the combinatorial explosion caused by the greater queue length.

Therefore, the effect of adding attributes to a queuing system is not so much dependent on the original state space of the system, but rather on the possible maximum number of customers in the system, which is causes the number of possible combinations of customers attributes

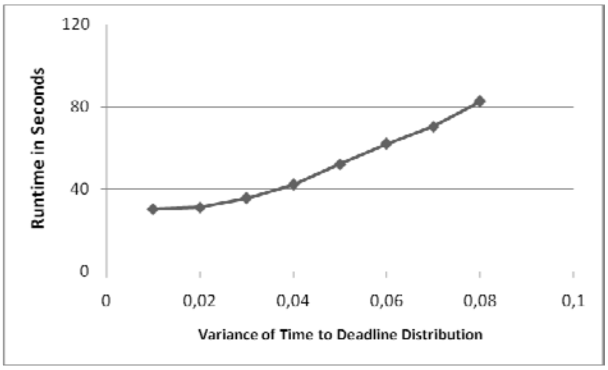


Figure 6: Runtime for Different Deadline Distributions

The last experiment shows the effect of different deadline distributions on the computation cost of the method. The queuing system used has normally distributed interarrival times (~N(0.7,0.2)) and normally distributed service times (~N(0.5,0.1)). Figure 6 shows the development of the algorithm runtime using different deadline distributions with a mean of 0.5 and with the number of possible time-to-deadline values varying from 2 to 12.

The initial simulation runtime of 30 seconds is already large considering a maximum possible queue length of 5, but the runtime increase for supposedly larger value set is not that severe. This happens because the value of the attribute time to deadline is not static. In the course of the simulation the value is updated by decreasing it in every time step, so that it reaches zero when the deadline approaches, afterwards it becomes negative. Therefore the actual value space of the attribute deadline ranges from the maximum distribution value to the longest delay and is larger than the initial number of discretized distribution values.

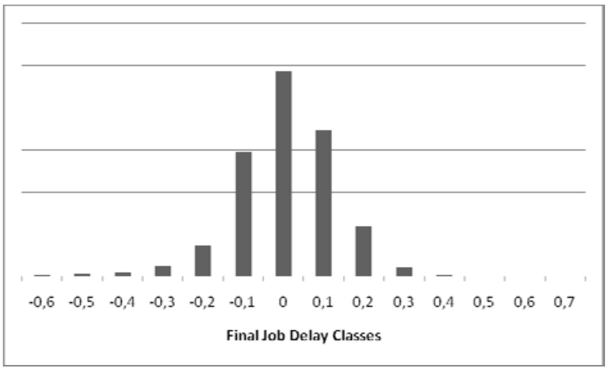


Figure 7: Throughput of Customers with Different Delays

In our experiment, the maximum delay was 1.7 minutes, which increases the maximum number of attribute values to almost 30 in the worst case. Figure 7 shows the throughput for the different final delays. The most frequent attribute values can be found around zero, as is expected when the service time and deadline distribution have the same mean. All other classes between -1.7 and 1.1 have much lower frequencies.

All experiments show a significant increase in memory and runtime cost due to adding attributes, as can be expected. However, clever storage and coding strategies help reduce these costs. The number of different attribute values that can be added feasibly differs between the attributes. The maximum number of priority classes is about 10, also depending on the maximum possible number of customers in the system. The discrete time steps of a distribution can be up to 20 or more, when the extreme values are less frequent than the average ones, such as when using a normal distribution. This happens because the simulation algorithm truncates Proxels below a certain probability threshold to limit the computation complexity.

Overall, the inclusion of attributes into Proxel-based queuing system simulation is feasible for attributes with a small value set or a small number of frequent attribute values.

5. CONCLUSION AND OUTLOOK

The paper described how Proxel-based queuing system simulation can be improved by adding attributes to the customers. This increases the number of possible applications and realism of the simulated queuing system classes. Limiting the approach to only one static attribute was necessary to keep the model's size in check. Furthermore, efficient storage strategies are essential for handling of the state space explosion.

Adding attributes leads to an increase in computation cost. This currently limits the applicability to queuing systems with general distribution functions with small variances or a low mean value. Adding one attribute with a small value set is possible.

The developed approach can be useful to queuing analysts, if no analytical solution is available for a queuing system. Especially rough estimates of the performance measures can be obtained very fast. Interesting results can be obtained such as the distribution of waiting time over the possible values of a customers processing time or the probability for infinite postponement.

Since we were able to implement the attribute processing time at a low cost, one future task is to implement round-robin as a queuing strategy. Implementing multiple attributes per customer is possible, but will only be feasible in conjunction with even more efficient storage strategies.

REFERENCES

Bolch, G., Greiner, S., de Meer, H., Trivedi, K.S., 2006. *Queueing Networks and Markov Chains*. John Wiley & Sons, Hoboken, New York, 2nd edition.

Casale, G., 2006. An efficient algorithm for the exact analysis of multiclass queueing networks with large population sizes. In *Performance Evaluation Review*, Vol. 34 (1), pp 169-180.

Cremonesi, P., Schweitzer, P. J., Serazzi, G., 2002. A Unifying Framework for the Approximate Solution of Closed Multiclass Queuing Networks. In *IEEE Transactions on Computers*, Vol. 51 (12), pp. 1423 - 1434.

Gross, D., Harris, C. M., 1998. *Fundamentals of Queueing Theory*. John Wiley & Sons, New York, 3rd edition.

Hasslinger, G., Kempken, S., 2006. Transient analysis of a single server system in a compact state space. In *Proceedings of 13th International Conference on Analytical and Stochastic Modelling Techniques and Applications*. European Council for Modelling and Simulation.

Heindl, A., 2001. Decomposition of General Tandem Queueing Networks with MMPP Input. In Performance Evaluation, vol. 44 (1-4), pp 5-23.Krull, C., Horton, G., 2007. Application of Proxels to Queuing Simulation. In *Proceedings of Simulation and Visualization 2007*, Magdeburg, Germany.

Horton, G., 2002. A new paradigm for the numerical simulation of stochastic petri nets with general firing times. In *Proceedings of the European Simulation Symposium 2002*. SCS European Publishing House.

Lazarova-Molnar, S. 2005. *The Proxel-Based Method: Formalisation, Analysis and Applications*. PhD thesis, Otto-von-Guericke-University Magdeburg. Germany.

Whitt, W., 1994. Towards better multi-class parametric-decomposition approximations for open queueing networks. In the *Annals of Operations Research*, Springer Netherlands, Vol. 48 (3), pp. 221-248.

Xu, W. 2008. *Application of Proxels to Queuing Simulation with Attributed Jobs*. Masters thesis, Computer Science Department, Otto-von-Guericke-University Magdeburg, Germany.

AUTHORS BIOGRAPHY

CLAUDIA KRULL studied Computer Science at the Otto-von-Guericke-University Magdeburg, obtaining her Diploma in 2003. She spent an exchange year at the University of Wisconsin, Stevens Point, where she graduated in 2002. In April 2008 she successfully defended her PhD thesis at the Otto-von-Guericke-University Magdeburg.

GRAHAM HORTON studied Computer Science at the University of Erlangen, obtaining his Masters degree ("Diplom") in 1989. He obtained his PhD in Computer Science in 1991 and his "Habilitation" in 1998 at the same university, in the field of simulation. Since 2001, he is Professor for Simulation and Modelling at the Computer Science department of the University of Magdeburg.

SIMULATION AND OPTIMIZATION OF VEHICULAR FLOWS IN A HARBOUR

Raffaella Frattaruolo[(a)], Rosanna Manzo[(b)], Luigi Rarità[(c)]

Department of Information Engineering and Applied Mathematics,
University of Salerno, Via Ponte Don Melillo, 84084, Fisciano (SA), Italy

[(a)]frattaruolo@diima.unisa.it, [(b)]manzo@diima.unisa.it, [(c)]lrarita@unisa.it

ABSTRACT

This paper focuses on the simulation and optimization of car traffic, modelled through a fluid dynamic model, based on the conservation of the cars number. In particular, the case study of the harbour of Salerno, Italy, is presented. Simulations of vehicular flows are carried out by a graphical tool, that allows to reproduce the evolution of densities on the roads of the network through animated and coloured pictures. From the analysis of simulated densities on roads, obtained by a given input configuration for the network object of study, the tool is able to plan some strategies for the improvement of traffic conditions through an optimization routine.

Keywords: road networks, simulation, optimization.

1. INTRODUCTION

The aim of this paper is to present some simulations of the car traffic dynamics in the harbour of Salerno, Italy, and to consider an optimization study of vehicular flows in order to improve traffic conditions. The choice of the considered urban network, that belongs to the harbour of Salerno, is due to the fact that it is a critical point, since it separates the centre of the city from the highway. Hence, the harbour is interested either from a heavy car traffic, mainly coming from the highway, or from trucks, daily crossing around the harbour areas.

The simulations of urban traffic flows have been obtained using a numerical tool, based on a fluid dynamic model for car traffic introduced in Coclite et al. 2005. In last years, fluid dynamic approaches have been often used for several reasons. For example, they are evolutive, allowing the description of a given physical phenomenon in every instant of time; they are characterized by few parameters, and this permits simulations based on numerical schemes that are, at the same time, fast and accurate, with consequent study of networks of big dimensions. Moreover, as such models are based on conservation laws, according to which a given quantity has to be conserved, they have a wide range of applications: road and telecommunication networks, supply chains, gas pipelines, irrigation channels, etc. In particular, the model for road networks, that we consider, is based on the conservation laws formulation proposed by Lighthill and Whitham (Lighthill et al. 1955) and Richards (Richards 1956). On each single road, the traffic evolution satisfies the conservation law:

$$\partial_t \rho + \partial_x f(\rho) = 0, \qquad (1)$$

where $\rho = \rho(t,x) \in [0, \rho_{\max}]$, $(t,x) \in \mathbb{R}^2$, is the density of cars, $\rho_{\max}$ is its maximal value, $f(\rho) = \rho v(\rho)$ is the flux and $v(\rho)$ the average velocity. In what follows, we assume that the velocity is a smooth decreasing function of the density, and that f is concave. A similar approach can be very useful for the description of phenomena, such as shock formation and their propagation (Bressan 2000, Dafermos 1999), although such formulation is limited for the reconstruction of some real traffic characteristics on highways, for which more rich models are necessary (Bellomo et al. 2005, Kerner 2004).

Recently, these approaches, of fluid dynamic type, have been extended to urban networks (for an exhaustive presentation, see Garavello et al. 2006b and for the relative numerical schemes Bretti et al. 2006): some are based on the LWR model (Cascone et al. 2007, Cascone et al. 2008, Godunov 1959, Helbing et al. 2005), other on the second order model by Aw – Rascle (Garavello et al. 2006a).

The greatest part of papers deals with results obtained by the LWR model because, if we consider the evolution of traffic in urban context, some particular situations (that are not present on highways, for example) can occur, for example short roads, traffic interruptions, reduced velocities, and so on, which can be captured by the LWR model.

Also optimization problems for traffic flows were addressed, in order to find the correct choice of traffic parameters for avoiding congestion phenomena and/or improving car traffic circulation: Helbing et al. 2005 is devoted to traffic light regulation, while the works of Gugat et al. 2005, Herty et al. 2003, are more related to our analysis but focus on the case of smooth solutions (not developing shocks) and boundary control.

In Cascone et al. 2007, Cascone et al. 2008, two cost functionals have been introduced to measure the traffic behaviour. The first functional J_1 measures the average velocity of drivers on the network, while the second J_2 measures the expected mean travelling time on the network. The optimization is done over right of way parameters and traffic distribution coefficients.

Starting from the model of networks with LWR and using the optimization procedure developed in Cascone et al. 2007, Cascone et al. 2008, it was possible to build a numerical software for the evolution of car traffic, which is useful in order to make some prevision of real conditions of vehicular behaviour and to indicate the best interventions on the network (in terms of traffic lights, and signals) in order to improve traffic conditions. The tool for urban traffic elaborates animated and coloured graphics to let the traffic analysis easier for users. The goodness of simulation and optimization algorithms is tested on the case study of the harbour of Salerno, via simulations.

The paper is organized as follows. First, we give some basics about the model for car traffic and the optimization techniques and numerical schemes useful to approximate the equations of the model. Then, we consider the structure of the graphical tool and present some simulation and optimization results for the harbour of the city of Salerno, Italy.

2. ROAD NETWORKS MODEL AND OPTIMIZATION TECHNIQUES

A road network consists of a finite set of roads, modelled by intervals $I_i = [a_i, b_i] \subset \mathbb{R}$, $i = 1, ..., N$, $a_i < b_i$, with one of the endpoints that can be infinite. Roads are connected to some junctions, and each junction J has a finite number of incoming and outgoing roads; hence, the complete model is described by the couple (I, J), where $I = \{I_i \,:\, i = 1, ..., N\}$ is the set of roads, while J is the set of junctions. On each road, the evolution is given by (1). We consider a linear decreasing velocity:

$$v(\rho) = v_{\max}\left(1 - \frac{\rho}{\rho_{\max}}\right), \tag{2}$$

where $v_{\max}$ represent the maximal velocity of cars. The flux is given by:

$$f(\rho) = v_{\max}\rho\left(1 - \frac{\rho}{\rho_{\max}}\right). \tag{3}$$

Without loss of generality, from now on, we can consider $v_{\max} = \rho_{\max} = 1$. Consider now a junction J with n incoming roads and m outgoing roads. For simplicity, assume that I_i, $i = 1, ..., n$, are the incoming roads and I_j, $j = n+1, ..., n+m$, are the outgoing roads.

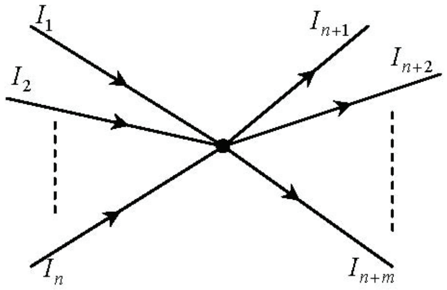


Figure 1: a road junction.

If $\rho = (\rho_1, ..., \rho_{n+m})$, $\rho_i \in [0, +\infty] \times I_i$, is a weak solution (Bressan 2000) at the junction such that, for each $x \mapsto \rho_i(t, x)$ has bounded variation, then ρ satisfies the Rankine – Hugoniot relation at the junction J:

$$\sum_{i=1}^{n} f\left(\rho_i\left(t, b_i^+\right)\right) = \sum_{j=n+1}^{n+m} f\left(\rho_j\left(t, a_j^-\right)\right), \tag{4}$$

for almost $t > 0$. Notice that (4) expresses precisely the conservation of cars through the junction.

For a single conservation law (1), a Riemann Problem (shortly RP) is a Cauchy problem for a piecewise constant data with only one discontinuity. Solutions are either continuous waves, called rarefactions, or traveling discontinuities, called shocks. The velocity of waves is strictly connected to $f'(\rho)$. In a similar way, we call RP at the junction the Cauchy problem corresponding to a constant initial data on each road. As the condition (4) does not guarantee the uniqueness of solutions, further conditions are needed.

The aim is finding a way to solve RPs at junctions.

Definition. A Riemann solver for the junction J is a map $RS : [0,1]^n \times [0,1]^m \to [0,1]^n \times [0,1]^m$ that associates to a Riemann data $\rho_0 = (\rho_{1,0}, ..., \rho_{n+m,0})$ at J a vector $\hat{\rho} = (\hat{\rho}_1, ..., \hat{\rho}_{n+m})$ and the solution on an incoming road $I_i, i = 1, ..., n$, is given by the wave $(\rho_{i,0}, \hat{\rho}_i)$ and on an outgoing road I_j, $j = n+1, ..., n+m$ is given by the wave $(\hat{\rho}_j, \rho_{j,0})$. The consistency condition is required: $RS(RS(\rho_0)) = RS(\rho_0)$.

For the traffic flow, in Coclite et al. 2005, in the case $m \geq n$ a Riemann solver, based on the following rules, was introduced:

(A) there are some fixed coefficients, that represent the preferences of drivers. Such coefficients indicate the distribution of traffic from incoming to outgoing roads, and they can be kept in a traffic distribution matrix:

$$A = \left\{\alpha_{ji}\right\}_{j=n+1,\dots,n+m,\ i=1,\dots,n} \in \mathbb{R}^{m\times n}, \tag{5}$$

such that

$$0 < \alpha_{ji} < 1,\ \sum_{j=n+1}^{n+m} \alpha_{ji} = 1, \tag{6}$$

for $i = 1,\dots,n$ and $j = n+1,\dots,n+m$, where α_{ji} is the percentage of drivers who, arriving from the i - th incoming road, direct to the j - th outgoing road.

Remark. If we refer to junctions with one incoming road ($n = 1$), *a*, and two outgoing roads ($m = 2$), *b* and *c*, respectively, then matrix *A* reduces to the column vector with components α and $1-\alpha$, where α (resp. $1-\alpha$) represents the probability that drivers could go the outgoing road *b* (resp. *c*), from the incoming road *a*.

(B) Respecting the rule (A), drivers behave such as the flux through J can be maximized.

If $m < n$, a yielding rule is needed. For example, if we consider a road junction with two incoming roads ($n = 2$), a and b, and one outgoing road ($m = 1$), *c*, we need a *right of way* parameter $p \in \left]0,1\right[$, and the yielding rule can be stated as follows:

(C) Assume that not all cars can enter the outgoing road and let C be the amount that can do it. Then, pC cars come from the first incoming road and $(1-p)C$ cars from the second one.

For further details, see Coclite et al. 2005 and Garavello et al. 2006.

In order to measure the efficiency of the network and hence obtain optimization results, in Cascone et al. 2007, Cascone et al. 2008, two cost functionals, measuring, respectively, the velocity at which cars travel through the network and the time taken by cars to travel on the network, were considered.

Since the model considers macroscopic quantities, the averages were estimated integrating over time and space the average velocity and the reciprocal of the average velocity, respectively. If ρ_i represents the density on road *i*, the following functionals were defined:

$$J_1(t) = \sum_i \int_{I_i} v\left(\rho_i(t,x)\right) dx, \tag{7}$$

$$J_2(t) = \sum_i \int_{I_i} \frac{1}{v\left(\rho_i(t,x)\right)} dx. \tag{8}$$

A fixed temporal interval $\left[0,T\right]$, for some $T > 0$, was considered.

For the regulation of traffic, the aim was to maximize J_1 and to minimize J_2. In this way, optimal parameters for regulating the behaviour of car traffic on networks were obtained.

Unfortunately, the analytical treatment of J_1 and J_2 for a complex network is very difficult, thus the following strategy, consisting of three steps, was adopted:

Step 1. Compute the optimal parameters for simple networks formed by a single junction and every initial data. For this, consider the asymptotic solution over the network (assuming infinite length roads so to avoid boundary data effects).

Step 2. For a complex network, use the (locally) optimal parameters at every junction, updating the value of the parameters at every time instant using the actual density on roads near the junction.

Step 3. Verify the performance of the (locally) optimal parameters via simulations.

The first step happens to be an hard task even for simple junctions. The reason for this is the hybrid nature of the problem, where continuous, time and space varying variables as ρ influence and are influenced by discrete variables as right of way parameters and traffic distribution coefficients. Thus, two special cases were treated: the 2×1 case with two entering and one exiting road; and the 1×2 case with one entering and two exiting roads. For the first type of junction, one has only one right of way parameter, called p. The second type of junction has no right of way parameter and only one traffic distribution coefficient α. It is possible to prove that, for the flux function that is considered here, $f(\rho) = \rho(1-\rho)$, the optimal solutions for p and α are the same either for J_1 or for J_2. For a systematic presentation of the optimization algorithms and obtained results for such cost functionals, see Cascone et al. 2007, Cascone et al. 2008.

The implementation of Step 2 is done for the case study of the harbour of Salerno. For Step 3, instead, we consider two different choices for right of way parameters and distribution coefficients: (locally) optimal (obtained through optimization algorithms), and fixed, assigned by the user on the basis of observations on the real network. Such choices are useful to test the goodness of the obtained optimization results.

3. A FINITE DIFFERENCE SCHEME FOR THE APPROXIMATION

The described mathematical model must be treated numerically in order to realize the tool for car traffic able to elaborate the densities for roads. We can refer to the finite difference method of Godunov (see Godunov 1959).

We define a numerical grid in $(0,T)\times\mathbb{R}^L$ according to the following notation:

- Δx is the space grid size;
- Δt is the time grid size;
- $(t_h,\, x_m)=(h\Delta t,\, m\Delta x)$ for $h\in\mathbb{N}$ and $m\in\mathbb{Z}$ are the grid points.

Consider the hyperbolic equation

$$\rho_t+f(\rho)_x=0,\ \ x\in\mathbb{R},\ t\in[0,T], \tag{9}$$

with initial data

$$\rho(x,0)=\rho_0(x). \tag{10}$$

A solution of the problems is constructed taking a piecewise constant approximation of the initial data, v_0^Δ. We set

$$v_m^0=\frac{1}{\Delta x}\int_{x_m}^{x_{m+1}}\rho_0(x)\,dx,\ \ m\in\mathbb{Z}\ , \tag{11}$$

and the scheme defines v_m^h recursively, starting from v_m^0.

Notice that waves in two neighbouring cells do not interact before time Δt if the CFL condition holds:

$$\Delta t\sup_{m,h}\left\{\sup_{u\in I\left(u_m^h,u_{m+1}^h\right)}\left|f'(u)\right|\right\}\le\frac{1}{2}\Delta x. \tag{12}$$

Then, we define the projection of the exact solution on a piecewise constant function

$$v_m^{h+1}=\frac{1}{\Delta x}\int_{x_m}^{x_{m+1}}v^\Delta(x,\, t_{h+1})\,dx. \tag{13}$$

Under the CFL condition, the solutions are locally given by the Riemann Problems and, in particular, the flux in $x=x_m$ for $t\in(t_h,\, t_{h+1})$ is given by

$$f(\rho(t,x_m))=f\left(W_R\left(0;\ v_{m-1}^h,\ v_m{}^h\right)\right), \tag{14}$$

where $W_R\left(\frac{x}{t};\ v_-,\ v_+\right)$ is the self – similar solution among v_- and v_+. As the flux is time invariant and continuous, setting $g^G(u,v)=f\left(W_R(0;\ u,v)\right)$ under the CFL condition, the scheme can be written as:

$$v_m^{h+1}=v_m^h-\frac{\Delta t}{\Delta x}\left(g^G\left(v_m^h,v_{m+1}^h\right)-g^G\left(v_{m-1}^h,v_m^h\right)\right). \tag{15}$$

The numerical flux g^G, for the flux we are considering, has the expression:

$$g^G(u,v)=\begin{cases}\min\left(f(u),f(v)\right), & \text{if}\ \ u\le v,\\ f(u), & \text{if}\ \ v<u<\sigma,\\ f(\sigma), & \text{if}\ \ v<\sigma<u,\\ f(v), & \text{if}\ \ \sigma<v<u,\end{cases} \tag{16}$$

where σ represents the value of ρ such that $f(\sigma)=\max_{\rho\in[0,\ \rho_{\max}]}f(\rho)$.

4. A TOOL FOR THE MANAGEMENT OF ROAD TRAFFIC

The tool for car traffic consists of three components:

- a web configuration component for the topological representation of the network, realized by a Java applet.
- The core of the application for the numerical methods, approximating conservation laws, that can be required by the Java applet.
- An intermediate layer, that allows the communication among the calculus core, the File System (necessary for saving informations) and web components.

We report the UML scheme of the application:

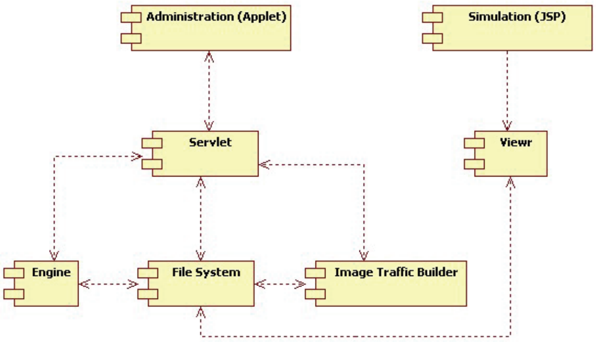


Figure 2: UML scheme.

We analyze the various components of the previous scheme.

File System
It is necessary for storing the various informations produced by the tool. In particular, inside the folder of the realized simulation project, three different folders are contained:

- the folder *InputFile* that contains .dat files, generated by the administration applet and used by the calculus engine for the elaboration of input informations.
- The folder *OutputFile* in which all .dbf files, that contain the numerical results of the simulation for every instant of time and produced as output of the engine, are stored.
- The folder *Images* with the pictures of the traffic state along the network as function of the corresponding time instant.

Administration Applet
It gives all the instruments to draw the topology of the network, with the insertion of all informations useful for the various arcs. Such information are stored in .dat files, contained in the File System in the folder *InputFile*. Such informations represent the input file for the calculus engine.

Engine
It is the numerical core for the application. It has access to the folder named *InputFile*, elaborates the .dat input files and produces .dbf files as output. Such files contain, for every instant of time, the densities for the various segments of each arc of the network to simulate. Such files are stored in the folder *OutputFile*.

Image Traffic Builder
It is the modulus that translates the numerical results (elaborated in .dbf files) of the folder *OutputFile* in images .jpg. Such images are stored in the folder *Images*.

Simulation JSP
It presents to the user all the saved projects for which animations are available (in such way that the user can refer to them if he wants).

Applet Viewer
The applet opens the files Images of the project associated to the simulation and executes in sequence the images of the simulation.

Servlet
This component allows to give to the applet all services available for building and simulating the network.

The tool for simulations is characterized by a simple user friendly interface. In particular, the use of mouse and some buttons allows to draw easily the network to analyze. For example, to draw arcs, it is necessary to click by the left button of the mouse at the center of the node and to direct the cursor toward the center of the node, that we want to connect (Figure 3).

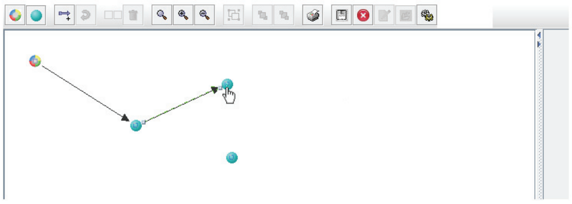
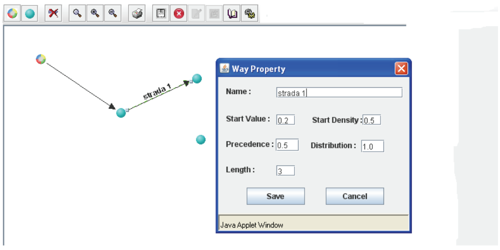


Figure 3: creation of an arc among two nodes.

For every arc and node, a menu is associated which is visualized clicking twice on them by the right button of the mouse. The menu for nodes consists of only one feature: *remove*, useful to remove the selected node. Instead, the menu for arcs is characterized by two features: *remove*, by which it is possible to clean the selected arc, and *properties*, by which it is possible to insert some characteristics for each arc (Figure 4). Such characteristics are the following:

- **Name:** name of the road;
- **Start Value:** initial density on the road;
- **Start Density:** incoming flux on the road (necessary as boundary data, if the node is virtual);
- **Precedence:** the right of way parameter (necessary if the road belongs to a junction with a number of incoming roads greater than the number of outgoing roads);
- **Distribution:** the distribution parameter, necessary to define traffic distribution matrix for the junction;
- **Length:** length of the road (in meters).



Name : strada 1
Start Value : 0.2
Start Density : 0.5
Precedence : 0.5
Distribution : 1.0
Length : 3

Figure 4: Edge property window.

After the simulation time has been chosen, the tool allows to reproduce the traffic evolution of the considered area. Different intervals of car densities are represented with different colours, as shown in the following table.

Table 1: Colours for densities.

Density interval	Colour
0.0 ≤ density < 0.1	■
0.1 ≤ density < 0.2	▒

$0.2 \leq$ density < 0.4	🟨
$0.4 \leq$ density < 0.6	🟩
$0.6 \leq$ density < 0.8	🟦
$0.8 \leq$ density ≤ 1.0	🟥

An optimization routine allows to set the characteristic parameters in such a way to avoid congestion phenomena.

5. THE CASE STUDY OF SALERNO HARBOUR

We present some simulation results for the harbour of Salerno, Italy. The network, built through the numerical software, is the following (Figure 5):

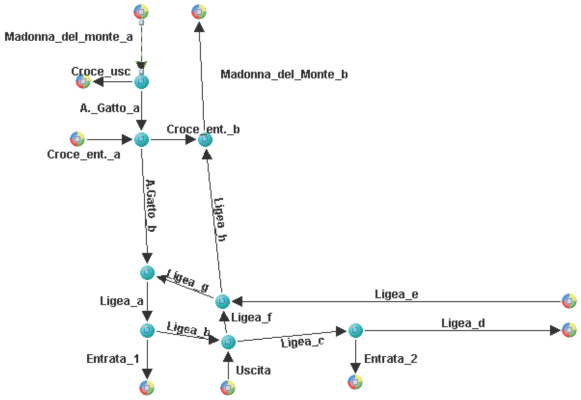


Figure 5: Network, that represents the harbour of Salerno.

In the following table, we show the input parameters for the network (notice that the incoming flux, that corresponds to the boundary data, is necessary only for roads connected to only one junction, which is to say roads with an infinite endpoint):

Table 2: Parameters for roads.

Name of the road	Initial density	Incoming flux	p	α
Madonna del monte a	0	0.6	1	1
Croce usc.	0	0	1	0.3
A. Gatto a	0	/	0.8	0.7
Croce ent. a	0	0.6	0.2	1
Croce ent. b	0	/	0.2	0.5
A.Gatto b	0	/	0.7	0.5
Ligea a	0	/	1	1
Ligea b	0	/	0.6	0.6
Entrata1	0	0	1	0.4
Uscita	0	0.3	0.4	1
Ligea c	0	/	1	0.6
Entrata 2	0	0	1	0.1
Ligea d	0	0	1	0.9
Ligea e	0	0.6	0.7	1
Ligea f	0	/	0.3	0.4
Ligea g	0	/	0.3	0.5
Ligea h	0	/	0.8	0.5
Madonna del monte b	0	0	1	1

These parameters are chosen by the user on the basis of the real networks observations and measures. We report the state of the considered vehicular flux in different instants of time. Such tool is very useful in order to detect queue formation on roads.

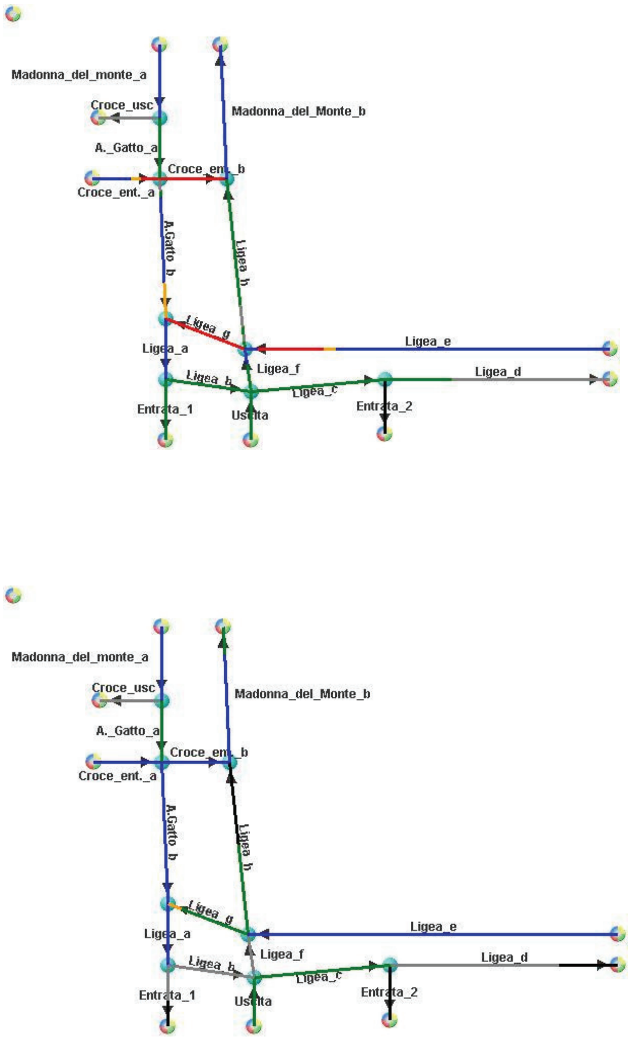


Figure 6: situation of the network at $t = 40$ (up) and $t = 60$ (down).

Let us refer to the Figure 6. At $t = 40$, we observe that, at the junction among *Via Gatto b* and *Via Ligea g* to *Ligea a*, there is a preliminary formation of queues along *Ligea g,* and this occurs because cars coming from these roads have not right of way with respect to cars coming from *A. Gatto b*.

It is interesting to notice how, as there are no cars that, from *via Ligea h*, have not reached the junction of *Croce ent b*, the flux coming from this last road is totally directed to *Madonna del monte b*.

At $t = 60$, the image underlines how the congestion along *Ligea g* is also propagating along *Ligea e*. In such instant, the flux of cars along *Ligea_h* has reached the junction of *Croce_ent_b*, and, on this last road, there is a queue formation due to a less right of way of the road itself with respect to *Ligea_h*.

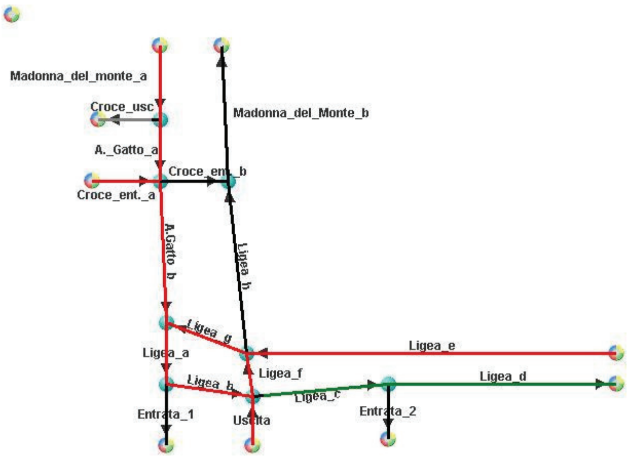


Figure 7: situation of the network at $t = 264$.

In the final instant of simulation (Figure 7, $t = 264$), there is a congestion phenomenon at the exit of the highway (*via Madonna del monte a*), and from the city center (*Ligea e*) to the junction among *via Ligea_e*, *Ligea f* and *Ligea g*. Moreover, the behaviour of the considered traffic is such that roads, that are directed to the highway, *Ligea h*, *Madonna del monte b* e *Croce ent b* are almost empty.

The simulative tool, on the basis of the network traffic characteristics, is able to suggest the user which are the more congested areas, on which it is suitable to make some interventions for the improvements of cars flows. The optimization procedure, once that a road junction, which presents congestions problems, is chosen, gives the optimal values of the characteristic parameters that can alleviate the congestion.

It is evident that queues and backward propagation occur on the road junction, characterized by two incoming roads, *A. Gatto b* and *Ligea g*, and one outgoing road, *Ligea a* (see Figure 8); hence, some optimization criteria are needed. In this case, the optimization of right of way parameters for the incoming roads is necessary.

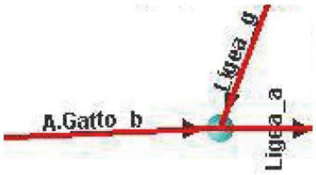


Figure 8: junction to optimize.

Let us give a meaning to the optimization outputs in terms of interventions on traffic signals or on traffic lights.

The modification of traffic signals corresponds, in the fluid dynamic model, to a different choice of traffic parameters, which regulate the behaviour of traffic at a road junction. For a better comprehension, we can refer to the following figure:

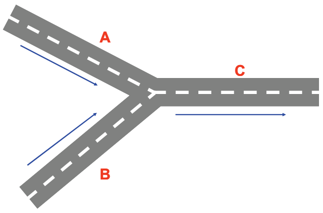


Figure 9: road junction with two incoming roads, *A* and *B*, and one outgoing road, *C*.

In this case, the regulation of traffic conditions for the road junction are the right of way parameters for the incoming roads *A* and *B*. In particular, if p is the right of way parameter for the road *A*, $1 - p$ is the corresponding one for the road *B*. Suppose that, for road *A*, a right of way parameter $p \in \left]0,\ 0.25\right[$ (and then for road *B* $q = 1 - p \in \left[0.75,\ 1\right[$) was obtained from the optimization procedure. The tool suggests that, to improve the traffic conditions, road *B* must have a higher right of way than road *A*. To obtain this, a stop signal for the road *A* can be introduced.

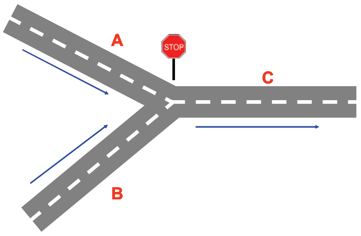


Figure 10: a STOP signal along road *A*.

Suppose that the optimal right of way parameter for road *A* is $p \in \left[0.25,\ 0.4\right]$. In this case, as road *B* has not a so high right of way as in the previous case, it is possible to insert a right of way signal for the road *A*, as in Figure:

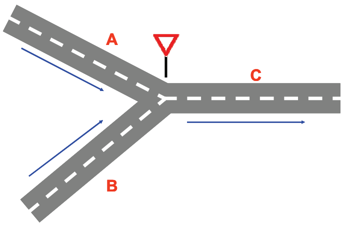


Figure 11: a yielding signal along road *A*.

Such traffic signals guarantee that cars, coming from the road *A*, will be able to stop if some cars are crossing the junction, coming from road *B*. Otherwise, they will be able to cross the junction, avoiding to necessarily stop, as in the previous case.

The optimization routine, applied to the chosen junction of Figure 8, indicates that a stop sign should be used on *Ligea g* and then some decongestion phenomena occur. The performances of the network improve, as indicated by Figure 12, that shows the behaviour of the functional J_1 in the optimal case and in the fixed case (the simulation defined by the user on the basis on real observations on the network). Notice that the primal aim, the maximization of the cars velocity, has been reached.

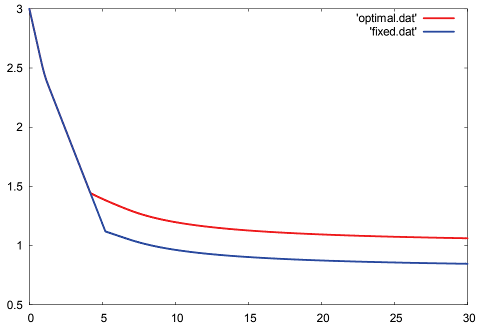


Figure 12: behaviour of the cost functional J_1 vs time in the optimal case (red) and in the fixed case (blue).

The insertion of traffic signals is surely the simplest intervention. However, in limit situations, determined by a high number of vehicles and frequent congestions, also in different hours of the day where it is necessary that drivers respect traffic rules, it is suitable to adopt traffic lights, which have to be adequately temporized.

Consider the road junction in Figure 13.

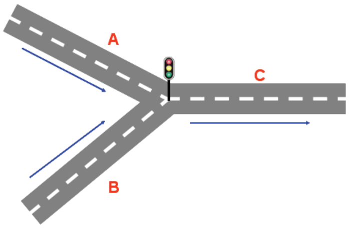


Figure 13: road junction with a traffic light.

The traffic light, in a generic instant of time, is red for a road and green for the other. In particular, if drivers for road *A* see the red phase, then drivers of road *B* can circulate. Hence, road *A* is characterized by a zero right of way parameter and, on the contrary, road *B* has a right of way parameter equal to 1. On the contrary, if drivers for road *A* see the green phase, they can circulate. Suppose that the optimization procedure establishes that, from the road *A*, a given percentage of traffic flux, *p*, should go the outgoing road. In this case, an adequate temporization of traffic light cycles is necessary. In particular, let Δ_v, Δ_r, and $T = \Delta_v + \Delta_r$, be, respectively, the green time, the red one, and the complete traffic light cycle. In this case, as *p* represents the percentage of drivers who, from road *A*, must cross the junction on average, then such parameter can be interpreted as the ratio among the green cycle and the total traffic light cycle. Hence, the road *B* is characterized by an averaged right of way parameter equal to $1-p$, or the ratio among the red cycle and the total cycle time. As a consequence, if *p* is the optimal right of way parameter, it is useful to design the traffic cycles such that $\Delta_v = pT$ e $\Delta_r = (1-p)T$.

In the previous example, for the optimization of car traffic either with traffic signals or with a traffic light, a road junction with two incoming roads and one outgoing road is considered. Such choice is not accidental. The typology of the presented junction is the most difficult to optimize, as it requires the modification of the parameter, that regulates the behaviour of car traffic at the junction, or the right of way parameter for a given road. The choice of such parameter is independent on the destination of drivers and, as a consequence, from paths that drivers choose to reach it.

The discussion made here for the optimization is valid for right of way parameters. As for traffic distribution coefficients, their optimization corresponds to some decisional criteria for the habits of drivers, that can change in cases of particular situations, as congestions. In such sense, distribution coefficients optimization allows the redirection of car flows to more free urban areas.

REFERENCES

Bellomo, N., Coscia, V., 2005. First order models and closure of the mass conservation equation in the mathematical theory of vehicular traffic flow, *CR Mechanique 333*, pp. 843-851.

Bressan, A., 2000. Hyperbolic Systems of Conservation Laws - The One - dimensional Cauchy Problem, Oxford Univ. Press.

Bretti, G., Natalini, R., Piccoli, B., 2006. Numerical approximations of a traffic flow model on networks, *Networks and Heterogeneous Media* 1, pp. 57-84.

Cascone, A., D'Apice, C., Piccoli, B., Rarità, L., 2007. Optimization of traffic on road networks, *Mathematical Models and Methods in Applied Sciences,* 17, n. 10, pp. 1587-1617.

Cascone, A., D'Apice, C., Piccoli, B., Rarità, L., 2008. Circulation of car traffic in congested urban areas, *Accepted for publication at CMS.*

Chitour, Y., Piccoli, B., 2005. Traffic circles and timing of traffic lights for cars flow, *Discrete and Continous Dynamical Systems - Series B* 5, pp. 599-630.

Coclite, G., Garavello, M., Piccoli B., 2005. Traffic Flow on Road Networks, *SIAM Journal on Mathematical Analysis* 36, pp. 1862-1886.

Dafermos C., 1999. Hyperbolic Conservation Laws in Continuum Physics, *Springer - Verlag.*

Garavello, M., Piccoli B., 2006. Traffic flow on a road network using the Aw – Rascle model, *Comm. Partial Differential Equations* 31, pp. 243-275.

Garavello, M., Piccoli B., 2006. Traffic flow on networks, *Applied Math Series* Vol. 1, American Institute of Mathematical Sciences.

Godlewsky, E., Raviart P., 1996. Numerical Approximation of Hyperbolic Systems of Conservation Laws, *Springer Verlag*, Heidelberg.

Godunov, S. K., 1959. A finite difference method for the numerical computation of discontinuous solutions of the equations of fluid dynamics, *Mat. Sb.* 47, pp. 271-290.

Gugat, M., Herty, K., Klar, A., Leugeuring, G., 2005. Optimal Control for Traffic Flow Networks, *Journal of Optimization Theory and Applications* 126, pp. 589-616.

Helbing, D., Lammer, S., Lebacque, J. P., 2005. Self-organized control of irregular or perturbed network traffic, *Optimal Control and Dynamic Games*, C. Deissenberg and R. F. Hartl eds., Springer, Dordrecht, pp. 239-274.

Herty, M., Klar, A., 2003. Modelling, Simulation and Optimization of Traffic Flow Networks, *SIAM J. Sci. Comp.* 25, pp. 1066-1087.

Herty, M., Moutari, S., Rascle, M., 2006. Optimization criteria for modelling intersections of vehicular traffic flow, *Networks and Heterogeneous Media* 1, pp. 275-294.

Holden, H., Risebro, N. H., 1995. A Mathematical Model of Traffic Flow on a Network of Unidirectional Roads, *SIAM J. Math. Anal.*, 26, pp. 999-1017.

Kerner, B., 2004. The Physics of Traffic, *Springer*.

Lebacque, J. P., 1996. The Godunov scheme and what it means for first order traffic flow models, *Proceedings of the Internaional symposium on transportation and traffic theory* 13, Lyon , Pergamon Press, Oxford, pp. 647-677.

Lighthill, M. J., Whitham, G. B., 1955. On kinetic waves. II. Theory of Traffic Flows on Long Crowded Roads, *Proc. Roy. Soc. London Ser. A*, 229, pp. 317-345.

Richards, P. I., 1956. Shock Waves on the Highway, Oper. Res., 4, pp. 42-51.

AUTHORS BIOGRAPHY

RAFFAELLA FRATTARUOLO was born in Foggia, Italy, in 1980. She graduated cum laude in Mathematics with a thesis on the numerical treatment of integral equations. She is actually a PhD student in Mathematics at the University of Salerno. Her scientific interests are about dynamic models for traffic flows on road and telecommunication networks.
Her e-mail address is frattaruolo@diima.unisa.it.

ROSANNA MANZO was born in Polla, Salerno, Italy. She graduated cum laude in Mathematics in 1996 and obtained PhD in Information Engineering in 2006. She is researcher at the Department of Information Engineering and Applied Mathematics of the University of Salerno. Her research areas include: fluid – dynamic models for traffic flows on road, telecommunication networks and supply chains, queueing theory, self – similar processes and computer aided learning.
Her e-mail address is manzo@diima.unisa.it.

LUIGI RARITÁ was born in Salerno, Italy, in 1981. He graduated cum laude in Electronic Engineering in 2004, with a thesis on mathematical models for telecommunication networks, in particular tandem queueing networks with negative customers and blocking. He obtained PhD in Information Engineering in 2008 at the University of Salerno. He is actually a research assistant at the University of Salerno. His scientific interests are about queueing theory, numerical schemes and optimization techniques for fluid – dynamic models.
His e-mail address is lrarita@unisa.it.

A GRAPHICAL TOOL FOR THE SIMULATION OF SUPPLY CHAINS USING FLUID DYNAMIC MODELS

Alfredo Cutolo[(a)], **Carmine De Nicola**[(b)], **Rosanna Manzo**[(c)]

Department of Information Engineering and Applied Mathematics,
University of Salerno, Via Ponte Don Melillo, 84084, Fisciano (SA), Italy

(a)cutolo@diima.unisa.it, (b)denicola@diima.unisa.it, (c)manzo@diima.unisa.it

ABSTRACT

This paper presents a numerical tool for the simulation of supply chains based on two different fluid dynamic models. One is based on a mixed continuum-discrete model, it means that the load dynamics are solved in a continuous way on the arcs, and at the nodes imposing the conservation of the goods density, but not of the processing rate. In fact, each arch is modelled by a system of two equations: a conservation law for the goods density, and an evolution equation for the productive capacity. According to the other model, the load dynamics are described by a conservation law, with constant processing rate, inside each supply sub-chain, and an entering queue for exceeding parts. The dynamics at a node are solved considering an ode for the queue. The realized tool allows to reproduce the state of the densities on arcs through coloured animations.

Keywords: conservation laws, supply chains, simulation.

1. INTRODUCTION

In last years, scientific communities showed a great interest for modelling the dynamics of industrial production, managed by supply chains. The study of such dynamics can become fundamental in order to reduce some unwished phenomena (bottlenecks, dead times, and so on), which can lead to heavy delays in production processes.

Several mathematical approaches have been proposed in order to model supply chains. For example, some models are discrete and based on considerations of individual parts. Other models are continuous (see Armbruster et al. 2006a, Armbruster et al. 2006b, Armbruster et al 2004, Daganzo 2003), and based on partial differential equations.

Probably, the first paper, that relies on continuous equations, was Armbruster et al. 2006a, where the authors, via a limit procedure on the number of parts and suppliers, have obtained a conservation law (see Bressan 2000, Dafermos 1999), whose flux involves either the parts density or the maximal productive capacity.

But other continuous models for supply chains have been introduced (see Bretti et al. 2007, D'Apice et al. 2006, Gottlich 2005, Gottlich 2006) due to the difficulty of finding a solution for the general equation proposed in Armbruster et al. 2006a. Also extensions on networks have been made (D'Apice et al. 2008, Helbing et al. 2004, Helbing et al. 2005).

In this paper, we focus the attention on two different continuous models for supply chains and networks (D'Apice et al. 2006, Gottlich et al. 2005) for the simulation of chains (and more complicated networks) of big dimensions.

In particular, the model introduced in D'Apice et al. 2006, briefly (DM) model, describes supply chains by continuous arcs and discrete nodes. This implies that the load dynamics are solved in a continuous way on the arcs by a conservation law for densities and a wave equation for the maximum processing rates, and at the nodes imposing the conservation of the goods density, but not of the processing rate. In fact, each arch is modelled by a system of two equations: a conservation law for the goods density, and an evolution equation for the productive capacity.

Instead, according to the model proposed in Gottlich et al. 2005, briefly (GHK) model, the load dynamics are described by a conservation law, with a constant processing rate inside each supply sub-chain. Moreover, the adoption of queues for modelling the transition of goods among arcs is proposed. Such choice allows an easy accessibility to existence and uniqueness of the solution to the whole network (Herty et al. 2007). Unlike the first model, for which a system of partial differential equations is considered, here we rely with ODEs for queues and conservation laws (PDEs) for densities on arcs.

It is evident that the described models complete each other. In fact, the second approach is more suitable when the presence of queues with buffers is fundamental to manage goods production. The mixed continuum-discrete model, on the other hand, is useful when there is the possibility to reorganize the supply chain: in particular, the productive capacity can be readapted for some contingent necessities.

Numerical schemes for the two models of supply chains have been developed to build the core of a graphical tool, that can be useful to simulate the behaviour of an assigned supply chain, in terms of parts densities on arcs. The tool is characterized by a graphical interface, which is user – friendly, since it contains a series of buttons for an easy construction of the supply network to simulate. Inside the work area, the user is able to visualize the effects of simulations through animated coloured pictures, which show the densities on the various arcs of the supply chain, and to evaluate the effects of changes in the supply chain organization. Of course, for the (GHK) model, it is possible also to see the evolution in time of queues at nodes.

The paper is organized as follows. First, we give some basics about the models for supply chains and numerical schemes useful to approximate the equations of the various models. Then, we consider the structure of the graphical tool and present some simulation results.

2. MATHEMATICAL MODEL

We describe briefly the two models for supply chains, separately. Let us consider the (GHK) model.

A supply chain (Gottlich et al. 2005, Gottlich et al. 2006) is characterized by a set of suppliers, connected each other, with the aim of processing goods, that travel along the arcs. Moreover, each supplier consists of a processor and a buffer, or queue.

Definition. A supply chain is a graph, consisting of a finite set of arcs, *J*, and a finite set of vertices, *V*. Each supplier *j* is modeled by an arc *j*, which is parameterized by an interval $\left[a_j, b_j\right]$.

Each processor *j* is characterized by a maximum processing rate μ_j, length L_j, and processing time T_j. The rate L_j / T_j describes the processing velocity. The dynamic of each processor on an arc *j* is governed by the following equation:

$$\partial_t \rho_j(x,t) + \partial_x \min\left\{\frac{L_j}{T_j}\rho_j(x,t), \mu_j\right\} = 0 \qquad (1)$$

$$\forall\, x \in \left[a_j, b_j\right],\ t \in \mathbb{R}^+,$$

$$\rho_j(x,0) = \rho_{j,0}(x)\ \ \forall\, x \in \left[a_j, b_j\right], \qquad (2)$$

where $\rho_j(x,t)$ is the density of parts in the processor *j* at the point *x* and at time *t*.

The equation (1) has the following interpretation: parts are processed with a velocity L_j / T_j, but with a maximum allowed flux μ_j. If the incoming flux, for the arc *j*, becomes higher than the maximal allowed flux μ_j, a queue is formed.

At first, we analyze the case in which every vertex of the network, consisting of *N* suppliers, is connected exactly to one incoming arc and one outgoing arc, assuming that arcs are labeled in sequence, or the arc *j* is connected to the arc $j+1$ and $b_j = a_{j+1}$. In this case, we are dealing with consecutive suppliers.

The queue associated to the supplier *j* is in front of each processor, or in $x = a_j$. Assume that the first supplier is characterized only by a processor with infinite length and that the last supplier has an infinite supplier, or $a_1 = -\infty$ and $b_N = +\infty$, respectively.

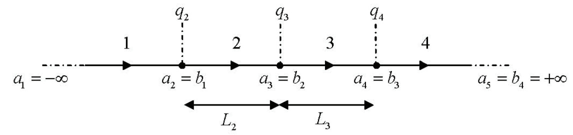


Figure 1: a scheme of a supply chain with $N = 4$.

The queue q_j for the processor *j* can be considered a function of time, or $q_j = q_j(t)$. Notice that there is no spatial dependence for the queue, as it is located at $x = a_j$. Variations of the length *j* are due to variations of fluxes for arcs $j-1$ and *j*. Precisely, the queue grows if the flux for the arc $j-1$ becomes higher than the flux for the arc *j*. Hence, the temporal variation of the queue, $\frac{d}{dt}q_j(t)$, satisfies the following differential equation:

$$\frac{d}{dt}q_j(t) = f_{j-1}\left(\rho_{j-1}\left(b_{j-1},t\right)\right) - f_j\left(\rho_j\left(a_j,t\right)\right), \qquad (3)$$
$$j = 2,\ldots,N.$$

If the second term of (3) is positive, $q_j(t)$ starts to increase. Notice also that the fluxes for the arcs $j-1$ and *j* are computed in the same spatial place that represents, as for the evolution of densities of arcs, a point where some discontinuities can occur.

The flux for the arc 1 of the supply chain is assigned and represents the incoming external flux. The remaining fluxes $f_j\left(\rho_j(x,t)\right)$, $j = 2,\ldots,N$, are given by:

$$f_j\left(\rho_j\left(a_j,t\right)\right) = \begin{cases} \min\left\{f_{j-1}\left(\rho_{j-1}\left(b_{j-1},t\right)\right), \mu_j\right\}, & q_j(t) = 0, \\ \mu_j,\ q_j(t) > 0. \end{cases} \qquad (4)$$

If the queue is empty, the flux of the arc $j-1$ can be processed on the arc j if such flux is lower than μ_j; otherwise, goods are processed at rate μ_j. If the queue is not empty, the flux $f_{j-1}\left(\rho_{j-1}\left(b_{j-1},t\right)\right)$ is higher than μ_j, hence goods are processed at rate μ_j.

Consider now the (DM) model. For exhaustive explanations and details, see D'Apice et al. 2006, D'Apice et al. 2008.

A supply chain consists of a sequence of $N+1$ sub-chains $I_1,\ldots, I_{N+1}$, and N suppliers or processors $P_1,\ldots,P_N$ with certain throughput times and capacities. The supplier P_k connects the sub-chain I_k to the sub-chain I_{k+1}.

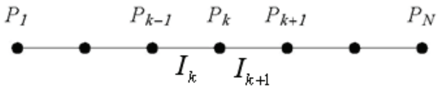


Figure 2: scheme of a supply chain.

Each supplier processes a certain good, measured in units of parts, and passes it in the next sub-chain. We assume that a node P_k consists of a processor, which decides how to manage the flow among sub-chains, with a maximal processing rate μ. Each sub-chain I_k is modelled by an interval $[a_k, b_k]$, with P_k corresponding to the coordinate b_k, on which we consider the system:

$$\begin{cases}\rho_t + f_\varepsilon\left(\rho,\mu\right)_x = 0,\\ \mu_t - \mu_x = 0.\end{cases} \tag{5}$$

For $\varepsilon > 0$, the flux function for each arc is defined in the following way:

$$f_\varepsilon^k\left(\rho,\mu\right) = \begin{cases}\rho, & 0 \le \rho \le \mu,\\ \mu + \varepsilon\left(\rho-\mu\right), & \mu \le \rho \le \mu_k^{\max},\end{cases} \tag{6}$$

or alternatively

$$f_\varepsilon^k\left(\rho,\mu\right) = \begin{cases}\varepsilon\rho + \left(1-\varepsilon\right)\mu, & 0 \le \mu \le \rho,\\ \mu + \varepsilon\left(\rho-\mu\right), & \rho \le \mu \le \mu_k^{\max},\end{cases} \tag{7}$$

with $\rho_k^{\max}$ and $\mu_k^{\max}$ the maximum density and processing rate on the arc I_k.

It is possible to generalize all following definitions and results to the case of different fluxes $f_{\varepsilon_k}^k$ for each line I_k (also choosing ε dependent on k). In fact, all statements are in terms of values of fluxes at endpoints of the sub-chains, thus it is sufficient that the ranges of fluxes intersect. Moreover, we can consider different slopes m_k for each line I_k, considering the following flux:

$$f_\varepsilon^k\left(\rho,\mu\right) = \begin{cases}m_k\rho, & 0 \le \rho \le \mu,\\ m_k\mu + \varepsilon\left(\rho-\mu\right), & \mu \le \rho \le \rho_k^{\max},\end{cases} \tag{8}$$

where $m_k \ge 0$ represents the velocity of each processor and is given by $m_k = \frac{L_k}{T_k}$, with L_k and T_k, respectively, fixed length and processing time of processor k.

We interpret the evolution at nodes P_k thinking to it as a Riemann problem (a Cauchy Problem corresponding to an initial data which is constant on each supply line) for the density equation with processing rates as parameters. Riemann problems are solved fixing different rules, which conserve the flux at nodes. Here, we refer to the following one:

(SC) The objects are processed in order to maximize the flux. Then, if a solution with only waves in the density ρ exists, then such solution is taken; otherwise, the minimal μ wave is produced.

Rule (SC) corresponds to the case in which processing rate adjustments are done only if necessary, while the density can be regulated more freely. Thus it is justified in all situations in which processing rate adjustments require re-building of the supply chain, while density adjustments are operated easily (e.g. by stocking).

Notice that algorithm (SC) is appropriate to reproduce also the well known "bull-whip" effect, a well known oscillation phenomenon in supply chain theory, see Daganzo 2003.

3. NUMERICAL METHODS FOR SUPPLY CHAINS

In order to simulate the behaviour of the supply chains, numerical schemes ad hoc have been realized in order to discretize the equations describing the mathematical models.

Let us refer to the (GHK) model. In this case, two different numerical methods have been used, one for the conservation law describing the evolution of the density and the other for the linear ordinary differential equation describing the evolution of the queue. For the conservation law, an upwind scheme is the more suitable because of the flux function shape of each arc. We deal with a discrete grid in the plane $\left(x,t\right)$ based on the finite difference method. A temporal step Δt and a spatial step h are chosen and the grid points $\left(x_j,t^n\right)$ are defined as follows:

$$\begin{aligned} &x_j = jh,\ j \in \mathbb{Z},\\ &t^n = n\Delta t,\ n \in \mathbb{N}.\end{aligned} \tag{9}$$

We set $\lambda = \Delta t / h$ (CFL condition, see Godunov 1959), and define $x_{j+1/2} = x_j + h/2$. The discrete solutions ρ_j^n, which approximate the density $\rho\left(x_j, t^n\right)$ for every *j* and *n*, for a generic arc of the supply chain, can be written as:

$$\rho_j^{n+1} = \rho_j^n - \lambda\left(H_{j+1/2}^n - H_{j-1/2}^n\right), \tag{10}$$

where *H* is the "numerical flux". For the upwind case, the numerical flux is equal to:

$$H_{j+1/2} = \frac{1}{2}\left[a\left(\rho_{j+1} + \rho_j\right) - a\left(\rho_{j+1} - \rho_j\right)\right], \tag{11}$$

where $a = \frac{L}{T}$, considering that *L* and *T* are referred to a generic arc, whose temporal evolution has to be computed.

In order to guarantee the convergence of the scheme, we choose $\left|\lambda \frac{L}{T}\right| \leq 1$.

The Euler method can be used for the queues equation (3). For the arc *j*, the approximation of the queue $q_j(t)$ at time t^{n+1}, $q_j\left(t^{n+1}\right)$, is q_j^{n+1}, which is found recursively as:

$$q_j^{n+1} = q_j^n + \Delta t\left(f_{j-1}^n - f_j^n\right), \tag{12}$$

where f_{j-1}^n and f_j^n are, respectively, the approximation of fluxes $f_{j-1}\left(\rho_{j-1}\left(b_{j-1}, t^n\right)\right)$ and $f_j\left(\rho_j\left(a_j, t^n\right)\right)$.

Initial data are necessary for every queue, and every arc. Moreover, a boundary condition, given by the input profile $f_1(t)$ is needed for each arc (see Gottlich et al. 2005, Gottlich et al. 2006). Such profile can be translated into initial data $\rho_{1,0}(x) := \rho_{1,0}\left(b_1 - t\right) = f_1(t)$ on the first arc (assumed artificial), with the assumptions that $\mu_1 > \max f_1$, and $\frac{L_1}{T_1} = 1$.

Consider now the (DM) model. In this case, we refer to a Godunov method for a 2×2 system (details are in Bretti et al. 2007, Godunov 1959). Let Δt and Δx be, respectively, the temporal and the spatial step for the discrete grid in the plane (x, t), whose points are $\left(x_j, t^n\right) = \left(j\Delta x, n\Delta t\right)$, $j \in \mathbb{Z}$, $n \in \mathbb{N}$; let ρ_j^n and μ_j^n be the approximations of $\rho\left(x_j, t^n\right)$ and $\mu\left(x_j, t^n\right)$, respectively. Then, the approximation scheme for (5) can be defined as follows:

$$\begin{cases} \rho_j^{n+1} = \rho_j^n - \dfrac{\Delta t}{\Delta x}\left(g\left(\rho_j^n, \rho_{j+1}^n\right) - g\left(\rho_{j-1}^n, \rho_j^n\right)\right), \\ \mu_j^{n+1} = \mu_j^n + \dfrac{\Delta t}{\Delta x}\left(\mu_{j+1}^n - \mu_j^n\right), \end{cases} \tag{13}$$

where the Godunov numerical flux *g* can be found solving Riemann problems among the states $\left(\rho_-, \mu_-\right)$ on the left and $\left(\rho_+, \mu_+\right)$ on the right:

$$g\left(\rho_-, \mu_-, \rho_+, \mu_+\right) = \\ = \begin{cases} \left(\rho_-, -\mu_+\right), & \textit{if } \rho_- < \mu_- \vee \rho_- \leq \mu_+, \\ \left(\dfrac{1-\varepsilon}{1+\varepsilon}\mu_+ + \dfrac{2\varepsilon}{1+\varepsilon}\rho_-, -\mu_+\right), & \textit{if } \rho_- < \mu_- \vee \rho_- > \mu_+, \\ \left(\dfrac{1+\varepsilon}{2}\rho_- + \dfrac{1-\varepsilon}{2}\mu_-, -\mu_+\right), & \textit{if } \rho_- \geq \mu_- \vee \mu_+ > \tilde{\mu}, \\ \left(\dfrac{1-\varepsilon}{1+\varepsilon}\left(\mu_+ + \varepsilon\mu_-\right) + \varepsilon\rho_-, -\mu_+\right), & \textit{if } \rho_- \geq \mu_- \vee \mu_+ \leq \tilde{\mu}, \end{cases} \tag{14}$$

with

$$\tilde{\mu} = \mu_- + \frac{1+\varepsilon}{2}\left(\rho_- - \mu_-\right). \tag{15}$$

For this numerical formulation of the model defined by (5), it is necessary to define the value of the boundary data, given by the term ρ_{j-1}^n. For the first arc of the supply chain, ρ_{j-1}^n is given by an input profile. If $m_1 = 1$, the boundary data is associated to the incoming external flux for the supply chain. Otherwise, ρ_{j-1}^n is determined by the solution to Riemann problems at nodes P_k.

Remark. The construction of the Godunov method is based on the exact solution to the Riemann problem (Bretti et al. 2007, Godunov 1959) in the cell $\left(x_{j-1}, x_j\right) \times (t^n, t^{n+1})$. To avoid the interaction of waves in two neighbouring cells before time Δt, we impose a CFL condition like:

$$\frac{\Delta t}{\Delta x}\max\left\{\left|\lambda_0, \lambda_1\right|\right\} \leq \frac{1}{2}, \tag{16}$$

where λ_0 and λ_1 are the eigenvalues of the system (5). Since, in this case, the eigenvalues are such that $\left|\lambda_0\right| = 1$, $\left|\lambda_1\right| \leq 1$, the CFL condition reads as: $\frac{\Delta t}{\Delta x} \leq \frac{1}{2}$.

4. WORKING AREA OF THE TOOL FOR SUPPLY CHAINS

The tool for supply chains, that can be used either for the (GHK) model or the (DM) model, consists of three not coupled components:

- a calculus component in C++, for the implementation of the numerical methods associated to the equations of the supply chains models;
- a graphical component, realized in Java, that communicate with the calculus component, in order to reconstruct the profiles of densities and the behaviour of queues through animations;
- a web component, which allows the tool to work under some web applications.

Notice that, using the graphical tool, the user has not to worry about the insertion of parameters for the numerical approximation (namely spatial and temporal steps for the numerical approximation discrete grid), since such parameters are automatically chosen by the tool in order to establish the convergence of numerical schemes.

In what follows, we describe how it is possible to create the graph of a supply chain and then to simulate the goods dynamics, using the (GHK) model. First of all, it is necessary to introduce the initial node. Notice that such node is virtual as it does not belong to the topology of the real network, but indicates the incoming point of the external profile of goods. Then, the user can insert the other (real) nodes of the supply chain, that has to be simulated. All these operations can be done through a series of buttons, that are in the upper part of the working area.

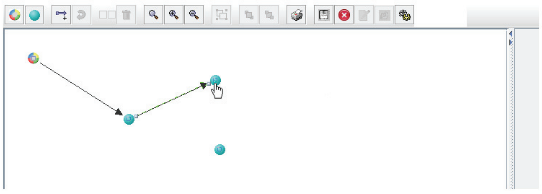
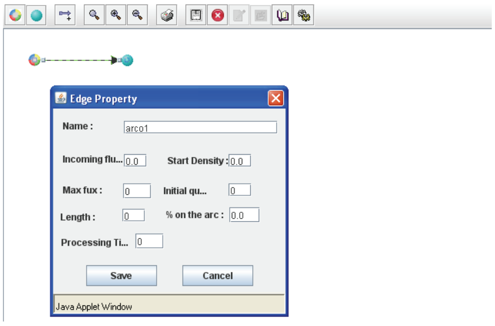


Figure 3: creation of an arc among two nodes.

For every arc and node, a menu is associated. The menu for nodes consists of only one feature: *remove*, useful to remove the selected node. The menu for the arcs is characterized by two features: *remove*, by which it is possible to clean the selected arc, and *properties*, by which it is possible to insert some characteristics for each arc (Figure 4):

- **Name:** name of the arc;
- **Incoming flux:** incoming flux (necessary as boundary data, although an external profile is always assigned);
- **Start Density:** initial density on the arc;
- **Max flux:** maximal flux for the arc;
- **Initial queue:** initial condition for the queue;
- **Length:** length of the arc;
- **Processing Time:** processing time;
- **% in the arc:** it indicates if the incoming flux can totally be directed to the outgoing arc; in supply chains, it is useful to choose it equal to one.



Edge Property

Name : arco1

Incoming flu... 0.0

Start Density : 0.0

Max fux : 0

Initial qu... 0

Length : 0

% on the arc : 0.0

Processing Ti... 0

Figure 4: edge property window.

After the creation of the graph, in order to start the simulation we have to choose the simulation time. Then, the evolution of the density on the arcs and the evolution of the queues can be visualized. Different colours are associated to every interval for parts densities along the arcs of the supply chain, according to the table below.

Table 1: table of colours.

Interval of density	Colour
0 $\leq$ density < 20	⬛
20 $\leq$ density < 40	🟩
40 $\leq$ density < 60	🟩
60 $\leq$ density < 80	🟦
80 $\leq$ density < 100	🟪
100 $\leq$ density < 120	🟨
120 $\leq$ density < 140	🟨
140 $\leq$ density < 160	🟧
160 $\leq$ density < 180	🟧
180 $\leq$ density $\leq$ 200	🟥

5. SIMULATION RESULTS

We present some simulation results for the supply chain in the following figure:

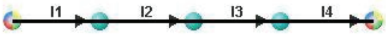


Figure 5: Supply chain with 4 arcs and 3 (real) nodes.

Let us first refer to the (GHK) model. Notice that the first arc is not artificial, as it does not contain any queue, unlike the other arcs. The considered simulation starts from an initial condition of empty network, with empty queue. The total simulation time is $T = 190$.

Length, maximal flux for each arc and processing time, or L_j, μ_j, e T_j, respectively, for $j = 1,2,3,4$, are kept in the following table, where details for the considered supply chains are reported:

Table 2: parameters for the supply chain.

Arc j	μ_j	T_j	L_j
1	25	1	1
2	15	3	2
3	10	4	3
4	15	2	1

We assume an input profile of triangular shape, as in Figure 6:

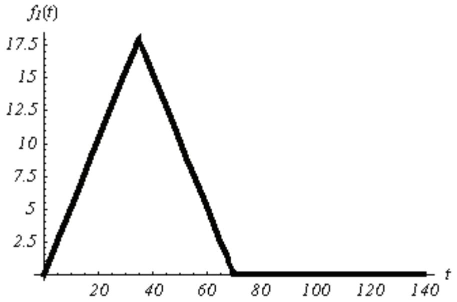


Figure 6: input profile for the firs arc of the supply chain.

In what follows, we report the state of goods traffic in some different instants of time for the analyzed supply chain, initially empty. Let us analyze Figure 7.

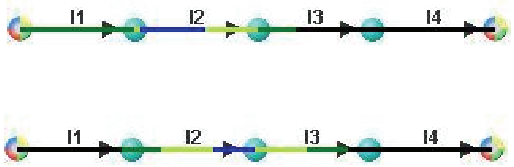


Figure 7: densities on arcs in time instants $t = 70$ (up) and $t = 90$ (bottom).

In Figure 7, at $t = 70$, the goods flux is quite low (it is evident from the colour of the first arc). A strong flux interests the second arc. The third arc is becoming to be full. At a first moment, the third arc is interested by a low goods flux. At $t = 90$, the readjustments of fluxes, because of queues at nodes, imposes a low flux on the first arc.

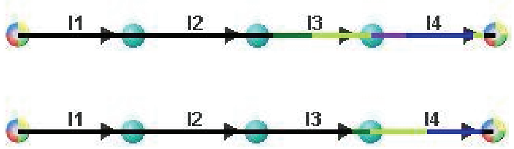


Figure 8: densities on arcs in time instants $t = 130$ (up) and $t = 160$ (bottom).

In Figure 8, at $t = 130$, the fourth arc is almost entirely full (violet and blue colours) and it is interested by a strong outgoing density. This is not surprising, since it means that there is a correct management of goods in the arcs 1, 2, and 3. Practically, the outgoing arc keeps all fluxes that have been re-elaborated by other arcs. At $t = 160$, the arc 4 starts to become empty.

In what follows, we consider also the temporal behaviour of queues, as you can see from Figure 9. In this case, the queue related to the arc four, $q_4(t)$ is zero, as $\mu_4 > \mu_3$. The different behaviours of queues $q_2(t)$ and $q_3(t)$ is essentially due to differences in processing velocity for arcs two and three. In particular, $v_2 = 0.66 < v_3 = 0.75$. We could expect that the arc three should process goods in a more faster way, but this does not occur since $L_3 > L_2$ and $T_3 > T_2$. This implies that, although $v_3 > v_2$, goods remain in the processor three for a greater time with respect to the processor two. Hence, $q_3(t)$ is higher than $q_2(t)$ and the needed time interval to let $q_3(t)$ become zero is higher than the necessary time interval for $q_2(t)$.

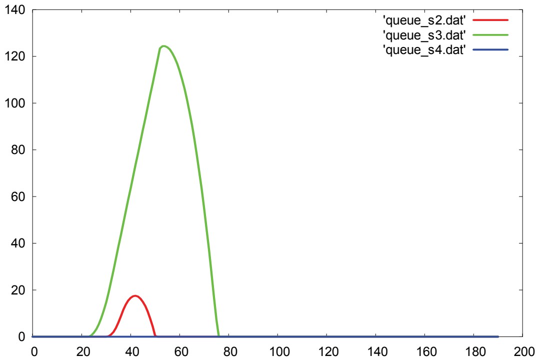


Figure 9: temporal behaviour for queues $q_2(t)$, $q_3(t)$ and $q_4(t)$, in red, green and blue, respectively.

Let us now to present some results obtained using the (DM) model. We consider a supply chain of $N = 4$ suppliers and impose the following initial and boundary data:

$$
\begin{aligned}
&\rho_1(0,x)=\rho_2(0,x)=\rho_3(0,x)=\rho_4(0,x)=0,\\
&\rho_1(t,0)=\frac{\mu_2}{2}\left(1+\sin\frac{3\pi t}{T_{\max}}\right),
\end{aligned}
\tag{17}
$$

where the space interval is [0,6] and the observation time is $T_{max} = 20$, with Δx=0.1 and Δt=0.05. On each processor, we assume that $\mu(0,x)=\mu_k$ and incoming and outgoing boundary data are given by μ_k. Notice that in this case the input profile $\rho_1(t,0)$ exceeds the maximum capacity of the processors.

We make simulations setting parameters as in the following table, assuming default processing velocities on each processor, namely $m_k=\frac{L_k}{T_k}=1,\ k=1,2,3,4$, where L_k is the length of the sub-chain and T_k the processing time:

Table 3: parameters for the supply chain.

Processor *k*	μ_k	L_k
1	99	1
2	15	1
3	10	3
4	8	1

In the next pictures, the evolution in time of flux, density and processing rate have been obtained using the rule (SC) for $\varepsilon = 0.1$.

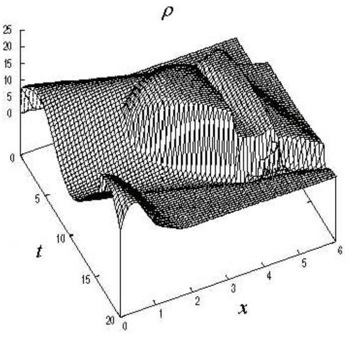


Figure 10: evolution of ρ.

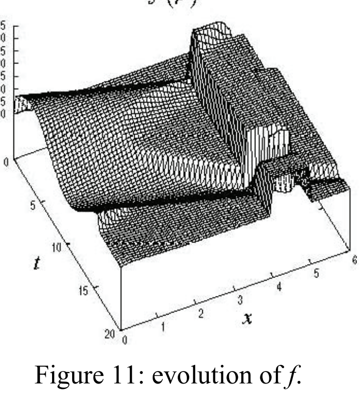


Figure 11: evolution of *f*.

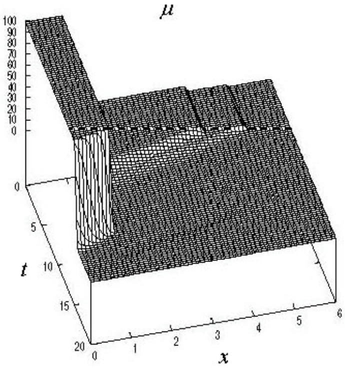


Figure 12: evolution of μ.

Since the algorithm (SC) maximizes the flux and adjusts the processing rate if necessary, minimizing its changes, we see that the processing rate of the first processor, initially equal to 99, in accordance to Table 3, is lowered to maximize the flux. The same happens with the other processors, even if the phenomena is less evident due to the initial values of the processing rate which are very low. Moreover we observer that the second processor is characterized by a great flow of goods.

From the analysis of the graphics we conclude that the algorithm (SC) gives rise to an interesting dynamics, which also fits with the (GHK) model. In particular, for ε tending to zero, the maximum values assumed by the flux and the density decrease.

CONCLUSIONS

In this paper, we described a tool able to simulate the behavior of the goods flow in a sequential supply chain using due different fluid dynamic models.

In future, we plan to integrate this tool with some optimization techniques in order to improve the performance of the supply in terms of a correct dimensioning of processing velocities and also of the processing rates for the (DM) model so as to reduce queues of production. Such study is highly not trivial, since one has to deal with the minimization of a numerical cost functional, which considers the lengths of queues in the case of the (GHK) model. This functional cannot be defined from an analytical point of view, as queues are evaluated numerically! Hence, the minimization problem requires the use of advanced numerical methods, which could be more expensive. The aim is elaborating fast algorithms, that could be able to achieve the required minimum with less computational times.

REFERENCES

Armbruster, D., Degond, P., Ringhofer, C., 2006. A model for the dynamics of large queueing networks and supply chains, *SIAM Journal on Applied Mathematics,* 66 (3), pp. 896-920.

Armbruster, D., Degond, P., Ringhofer, C., 2006. Kinetic and fluid models for supply chains supporting policy attributes, *Transportation Theory Statist. Phys.*

Armbruster, D., Marthaler, D., Ringhofer, C., 2004. Kinetic and fluid model hierarchies for supply chains, *SIAM J. on Multiscale Modeling,* 2 (1), pp. 43-61.

Bressan, A., 2000. Hyperbolic Systems of Conservation Laws - The One - dimensional Cauchy Problem, Oxford Univ. Press.

Bretti, G., D'Apice, C., Manzo, R., Piccoli, B., 2007. A continuum - discrete model for supply chains dynamics, *Networks and Heterogeneous Media (NHM),* 2 (4), pp. 661-694.

Dafermos C., 1999. Hyperbolic Conservation Laws in Continuum Physics, *Springer - Verlag*.

Daganzo, C., 2003. A Theory of Supply Chains, *Springer Verlag, New York, Berlin, Heidelberg*.

D'Apice, C., Manzo, R., 2006. A fluid dynamic model for supply chains, *Networks and Heterogeneous Media (NHM),* 1 (3), pp. 379-398.

D'Apice, C., Manzo, R., Piccoli, B., 2008. Modelling supply networks with partial differential equations, to appear in *Quaterly of Applied Mathematics*.

Helbing, D., Lammer, S., Seidel, P., Seba, T., Platkowski, T., 2004. Physics, stability and dynamics of supply networks, *Physical Review E* 70, 066116.

Helbing, D., Lammer, S., 2005. Supply and production networks: from the bullwhip effect to business cycles, in *D. Armbruster, A. S. Mikhailov, and K. Kaneko (eds.) Networks of Interacting Machines: Production Organization in Complex Industrial Systems and Biological Cells*, World Scientific, Singapore, pp. 33-66.

Godunov, S. K., 1959. A finite difference method for the numerical computation of discontinuous solutions of the equations of fluid dynamics, *Mat. Sb.* 47, pp. 271-290.

Gottlich, S., Herty, M., Klar, A., 2005. Network models for supply chains, *Communication in Mathematical Sciences*, 3(4), pp. 545-559.

Gottlich, S., Herty, M., Klar, A., 2006. Modelling and optimization of Supply Chains on Complex Networks, *Communication in Mathematical Sciences*, 4(2), pp. 315-330.

Herty, M., Klar, A., Piccoli, B., 2007. Existence of solutions for supply chain models based on partial differential equations, *SIAM J. Math. An.*, 39(1), pp. 160-173.

AUTHORS BIOGRAPHY

ALFREDO CUTOLO was born in Salerno, Italy, in 1980. He graduated in Informatics in 2005, with a thesis on a Geographical Information System (GIS). He is actually a PhD student in Mathematics at the University of Salerno. His scientific interests are about dynamic models for traffic flows on road and telecommunication networks.
His e-mail address is cutolo@diima.unisa.it.

CARMINE DE NICOLA was born in Salerno, Italy, in 1972. He graduated in Electronic Engineering in 2002 with a thesis on simulations of processor IAPX 86. He is actually a PhD student in Mathematics at the University of Salerno. His scientific interests are about fluid – dynamic models for the analysis of traffic flows on road and telecommunication networks.
His e-mail address is denicola@diima.unisa.it.

ROSANNA MANZO was born in Polla, Salerno, Italy. She graduated cum laude in Mathematics in 1996 and obtained PhD in Information Engineering in 2006. She is researcher at the Department of Information Engineering and Applied Mathematics of the University of Salerno. Her research areas include: fluid – dynamic models for traffic flows on road, telecommunication networks and supply chains, queueing theory, self – similar processes and computer aided learning
Her e-mail address is manzo@diima.unisa.it.

HEURISTIC PROCEDURES FOR PROBABILISTIC PROJECT SCHEDULING

Patrizia Beraldi [a], Maria Elena Bruni [b], Francesca Guerriero[c], Erika Pinto[d]

Dipartimento di Elettronica, Informatica e Sistemistica, University of Calabria,
Via P. Bucci 41C, 87030 Rende (Cosenza), Italy
[a] beraldi@unical.it, [b] mebruni@unical.it, [c]guerriero@unical.it[c]epinto@unical.it

ABSTRACT

In this paper we analyze the resource-constrained project scheduling problem under uncertainty. Project activities are assumed to have known deterministic renewable resource requirements and probabilistic activity durations described by random variables with a given density function. We develop heuristic algorithms for building a schedule with protected starting times, obtained using a buffering mechanism guided by probabilistic information.

Keywords: Stochastic project scheduling.

1. INTRODUCTION

The resource-constrained project scheduling problem (RCPSP) consists in minimizing the duration of a project, subject to the finish-start, zero-lag precedence constraints and the resource constraints. In its deterministic version the RCPSP, assumes complete information both on the resource usage and activities duration and determines a feasible baseline schedule, i.e. a list of activity starting times minimizing the makespan value. The role of the baseline schedule has been widely recognized in (Mehta and Uzsoy 1998; Möhring and Stork 2000), and it stems in supporting project decision makers providing a basis for planning external activities and facilitating resource allocation. Notwithstanding its importance, the planned baseline schedule in real contexts may have little, if some value, since project execution may be subject to severe uncertainty and then may undergo several types of disruptions as described in (Zhu, Bard and Yu 2005). In this paper we limit ourselves to represent uncertainty with stochastic activity durations. We shall refer to the insensitivity of planned activity start times to uncertain events as stability or solution robustness.

The stability of the program depends on what extent project managers consider uncertainty as a key features of project behaviors. In this paper we propose new heuristic procedures for generating a predictive schedule which exhibits acceptable solution robustness in the presence of multiple and frequent activity disruptions. We observe, that very few research papers explicitly consider probabilistic information in solution methods. We should mention here, the works (Van de Vonder, Demeulemeester and Herroelen 2008; Lambrechts, Demeulemeester and Herroelen 2008) for the case of uncertain resource availability. Besides this, our approach differs from the cited paper in some important aspects. In our paper we consider the stochastic programming framework and, in particular, the probabilistic paradigm in the form of joint probabilistic constraints. At the best of our knowledge none of the methods proposed in the literature consider joint probabilistic constraints. Even with some limitation, our scheduling approach can be tailored to reflect the level of risk that an individual decision maker is willing to bear in hedging against processing time uncertainty.

The remainder of the paper is organized as follows. In Section 2, we present a review of the relevant literature on project scheduling under uncertainty. In Section 3 we describe our scheduling methodology for generating robust baseline schedules. Section 4 is devoted to the presentation of the benchmark heuristics used to assess the efficacy of the newly developed heuristics and of the design of computational experiments. Results are analyzed in Section 5 and conclusions are presented in Section 6.

2. RELATED WORK

The methodologies for stochastic project scheduling basically view the project scheduling problem as a multistage decision process. Since the problem is rather involved and an optimal solution is unlikely to be found, scheduling policies are used for defining which activities to start at random decision points through time, based on the observed past and the a priori knowledge about the processing time distributions. (Igelmun and Radermacher 1983), propose a set of preselective scheduling policies. These policies, roughly speaking, define, for each possible resource conflict, a preselected activity that is postponed from a set of activities that cannot be executed together due to resource conflicts.

A branch-and-bound algorithm is developed in order to compute optimal preselective policies, and computational tests are reported for small instances. (Möhring and Stork 2000), introduce a new class of scheduling policies, called linear preselective policies intended to minimize the makespan for the RCPSP with stochastic activity durations. This new class combines the benefits of preselective policies and priority policies and is based on both determining sets of activities that cannot be executed together due to resource conflicts and on choosing an activity to be postponed according to a priority list. (Stork 2000), compares four scheduling policies for minimizing the makespan for the RCPSP with stochastic activity durations. Dominance rules and lower bounds are developed and then embedded into a branch-and-bound algorithm.

Research on heuristic procedures for solving the stochastic RCPSP is an active field of research. (Tsai and Gemmill 1998), propose a tabu search which makes use of a reduced neighborhood based on feasible swaps that can be executed on the current feasible sequence.

Given a feasible schedule, the makespan is computed by sampling repeatedly activity durations in order to obtainan estimate of the expected makespan. (Golenko- Ginzburg and Gonik 1997), develop a heuristic procedure operating in stages, where the decision to schedule the next activity is based on the precedence constraints and current resource availability.

A multiple knapsack problem minimizing expected project duration is proposed to solve resource conflicts. In a follow-up work, (Golenko-Ginzburg and Gonik 1998), apply a similar heuristic for a project scheduling problem where the duration of an activity is a random variable that depends on the amount of resources assigned to that activity. (Herroelen and Leus 2004), develop mathematical programming models for the generation of stable baseline schedules in a project environment without resource consumption.

They minimize the expected weighted sum of the absolute deviations between the planned and the actually realized activity starting times when exactly one activity duration disruption is expected to take place during project. (Tavares, Ferreira and Coelho 1998), study the risk of a project as a function of the uncertainty of the duration and the cost of each activity.

The authors make use of a buffering mechanism to increase the earliest activity start times. In (Rabbani, Fatemi, Ghomi, Jolai and Lahiji 2007), a newly developed resource-constrained project scheduling method in stochastic networks is presented which merges the critical chain concepts with traditional resource-constrained project scheduling methods.

The objective function takes into account the expected project duration and its variance. Since the developed model is a stochastic optimization model which cannot be solved in the general case, this paper suggests a heuristic algorithm where ready activities at each decision point are supplied by available resources on the basis of assigned priority level. These priority levels are the activities contribution in reducing the expected project duration and its variance. Therefore, the activities with the greatest probability to be on the critical chain and the greatest correlation with the project variance are fed-in first.

When resource availability constraints are considered, (Leus and Herroelen 2004), assuming the availability of a feasible baseline schedule, present exact and approximate formulations of the robust resource allocation problem, proposing for its solution a branch and-bound algorithm. The so-called resource flow network (Artigues and Roubellat 2000) is used to represent the flow of resources across the activities of the project network. In (Deblaere, Demeulemeester, Herroelen and Van de Vonder 2007) is presented a procedure for allocating resources to the activities of a given baseline schedule in order to maximize its stability in the presence of activity duration variability. The authors propose three integer programming based heuristics and one constructive procedure for resource allocation, thereby avoiding the use of stochastic variables. In (Van De Vonder, Demeulemeester, Herroelen and Leus 2006; Van de Vonder, Demeulemeester, Herroelen and Roel Leus 2005) a modi_cation of the ADFF heuristic presented in (Herroelen and Leus 2004; Herroelen and Leus 2004) is proposed in order to prohibit resource conflicts. In order to obtain a precedence and resource-feasible schedule, the resource flow-dependent float factor (RFDFF) heuristic uses information coming from the resource flow network in the calculation of the so called activity dependent oat factor. In (Van de Vonder, Demeulemeester and Herroelen 2008), multiple algorithms are introduced to include time buffers in a given schedule while a prede_ned project due date remains respected. While the virtual activity duration extension heuristic presented in (Van de Vonder, Demeulemeester and Herroelen 2008), relies on the standard deviation of the duration of an activity in order to compute a modified duration, the starting time criticality (STC) heuristic tries to combine information on activity weights and on the probability that activity cannot be started at its scheduled starting time.

Within the stochastic programming context, a two-stage integer linear stochastic model is proposed in (Zhu, Bard and Yu 2007) to determine target times in the first stage followed by the development of a detailed project schedule in the second stage with the aim of minimizing the cost of project completion and expected penalty incurred by deviating from the specified values.

Temporal protection is used against machine failure in (Gao 1995). The durations of activities requiring resources prone to breakdown are extended to provide extra time for protection. The protection equals the original duration augmented with the duration of breakdowns that are expected to occur during activity execution, based on breakdown statistics.

In (Lambrect, Demeleumester, Herroelen 2008) the case of uncertain resource availability due to a breakdown is tackled. The objective is to build a robust schedule that meets the project deadline and minimizes the schedule instability cost. In the paper it is shown that protection of the baseline schedule may provide significant performance gains over the use of deterministic scheduling approaches.

For an extensive review of research in this field, the reader is referred to (Herroelen and Leus 2004; Herroelen and Leus 2005).

3. SCHEDULING ACTIVITIES EXPLOITING PROBABILISTIC INFORMATION

The deterministic RCPS may be stated as follows. Let consider a project represented by a directed acyclic graph G = (N,V) and opt for activity on the node format (Wiest and Levy 1977). Each node in the set N corresponds to a single project activity and each arc in the set V corresponds to a precedence relation between each pair of activities. Each activity $j \in$ N, has to be processed without interruptions requiring a constant amount of resource r_{jk}, for each renewable resource type k, $k = 1..., K$. Each renewable resource is assumed to have a constant per period availability equal to a_k.

Let $S_t \subseteq$ V be the set of activities that are in progress at time period t. The RCPSP can be formulated as:

$$\min s_n$$

$$s_j \geq s_i + d_{\mathrm{i}} \quad (\mathrm{i, j}) \quad \mathrm{V} \tag{1}$$

$$\textstyle\sum_{i:i \in S_t} r_{ik} \geq \alpha_k$$

where d_i is the deterministic duration of activity i and s_i the planned starting time of activity i. Two kinds of constraints subsist among activities: precedence feasibility constraints which force activity j to be started only when all its immediate predecessor activities have been processed, and renewable resource constraints which prevent activities to exceed limited capacities of resources. The problem becomes even more involved in the stochastic RCPSP, where the durations of activities are not known in advance, but are instead represented by random variables $_j$ j N with known cumulative probability distribution function .

Given the uncertainty in activity duration, the decision maker may be inclined to solve several deterministic programs involving different values of the uncertain problem parameters or to replace the random variables by their expected value. As a matter of fact, the well known PERT model replaces randomness with a certainty equivalent in the form of expected value. However, none of these approaches can be considered satisfactory in face of uncertainty.

$$\min s_n \tag{2}$$

$$P(s_j \geq s_i + \qquad) \qquad (\mathrm{i, j}) \quad \mathrm{V} \tag{3}$$

$$\textstyle\sum_{i:i \in} \qquad \alpha_k \tag{4}$$

We observe that although the chance constraints paradigm has been tacitly accepted by the research community for over 30 years, recently, its validity as a tool for a point estimate of the project makespan or for the estimation of the complete cumulative density function of the makespan has been questioned. It is also well recognized that the chance constrained approach fails to give reasonable hints about the criticality of a path or activities, thereby destructuring critical chain approaches.

Indeed, the value taken by the starting time of an activity j, s_j is a function of a random variable and a decision variable s_i which is itself stochastic, its value depending on the starting times of preceding activities.

The appropriateness of a static chance constrained model, should at least be questioned. Nevertheless, a dynamic formulation in the form of a multistage recourse programming problem, may involve a huge number of scenarios, overwhelming form a computational point of view. These motivations are at the basis of the simplification of the problem on which our heuristics rely, based on a decoupling of the dynamic aspect of the problem from its probabilistic nature. In particular, the solution of the problem is viewed as a bilevel hierarchical process, where the temporal dependence is treated on the first level, whereas, stochasticity is introduced in a second level.

In the foregoing we shall present two different heuristics based on the general principle of joint probabilistic constraints.

3.1. The Joint Probabilistic Constraints heuristic

The Joint Probabilistic Constraints heuristic (JPCH) is inspired by heuristic algorithms for the deterministic RCPSP, but embeds the joint probabilistic constraints paradigm in its scheme. The schedule is constructed in two phases. In the first phase, a precedence feasible priority list is constructed following an ordering criterion. In the second phase, this priority list is transformed into a precedence and resource feasible schedule sequentially adding activities to the schedule until a feasible complete schedule is obtained.

In each step, g, or decision point t_g, the activities in the priority list, that are also part of the set E_g, containing all eligible activities which can be precedence and resource feasibly started at t_g, with better ranking are selected to be started at t_g and inserted in S_g. The set of already scheduled activities is denoted by A_g.

An algorithmic description of the Joint Probabilistic Constraints heuristic is given below:

- Initialization $g = 0$; $t_g = 0$;$A_0 = \{\emptyset\}$, $S_0 = \{\emptyset\}$.

- While $|A_g \cup S_g| \leq |N|$ do
  $g := g + 1$;
  $t_g = \min_{j \in Ag} c_j$;

- Calculate the residual resource availability and the set A_g and E_g.

- Use the priority rule to select the activities to be included in the set $S_g \subseteq E_g$.

```
•  Calculate the completion times of activities in $S_g$.
•  end while
```

Let us analyze in greater detail the problem of determining the completion times c_j of the set S_g of activities scheduled at decision point t_g, i.e., the starting times of activities to be scheduled at next decision point t_{g+1}. At each iteration g, the completion times are the solution of the following problem:

$$\min C \tag{5}$$
$$C \geq c_j \quad \forall j \in S_g \tag{6}$$
$$P\left(c_j \geq t_g + \hat{d}_j \quad \forall j \in S_g\right) \geq \alpha \tag{7}$$

where C is the next decision point t_{g+1}. In problem (5-7) the probabilistic constraints are jointly imposed on all the activities in S_g. This ensure that the probability of disrupting the starting time of successor activities is kept above the prescribed probability level α. As evident, this time buffering mechanism is used with the aim of absorbing potential disruptions caused by activity shifts. Buffer sizes are computed on the basis of the joint probability that activities in S_g disrupt subsequent activities. We observe that not considering joint probabilistic constraints would lead, at decision point t_g, to a probability disruption equal to the sum of the individual probability disruptions of activities in S_g.

The proposed heuristic, on the contrary, imposes a joint probability of disruption α for all the activities in S_g. We further remark that, considering the expected duration increase of activities i would result in disruption probability for each activity of at least 50% which can be unacceptable in some contexts.

3.2 The resource allocation heuristic

Rather than using a priority rule for deciding the set of activities to be included in S_g the resource allocation heuristic (RAH) determines at each decision point t_g, both the starting times and the resources allocated to each activity. Activities are not ordered in a list and if E_g contains more than one element a competition has to be arranged to choose the optimal subset of activities S_g that can be supplied by available resources. Decisions, at each t_g are made on the basis of the solution of the following subproblem.

$$\max \sum_{i \in E_g} \beta_i - \gamma * C$$

$$C \geq c_i \; \forall i \in E_g \mid \beta_i = 1$$

$$P\left(c_i \geq t_g + \hat{d}_i \; \forall (i) \in E_g \mid \beta_i = 1\right) \geq \alpha \tag{8}$$

$$\sum_{i: i \in E_g} r_{ik} * \beta_i \geq r_k(t_g) \; \forall k$$

where $r_k(t_g)$ is the residual resource availability at time t_g. The objective function tries to balance two conflicting objectives, the resource and the time allocation decisions. The two objective functions are weighted through the parameter γ, which is adjusted dynamically in order to find a good balance between conflicting objectives.

An algorithmic description of the resource allocation heuristic is given below:

```
•  Initialization $g = 0$; $t_g = 0$; $A_0 = \{\emptyset\}$, $S_0 = \{\emptyset\}$.
•  While $|A_g \cup S_g| \leq |N|$ do
•  g:=g+1.
•  $t_g = \min_{j \in A_g} c_j$ .
•  Calculate $E_g$, and update the residual resource availability and the set $A_g$. Set $\beta = 0{:}01$ $S_g = \{\emptyset\}$
•  While no activities have been selected ($S_g = \{\emptyset\}$):
       - Solve the competition problem (8). Let $\beta^*, C^*, c^*$ be the optimal solution.
       -If $\beta_i^* = 0$; $\forall i \in E_g$ $\beta := \beta/10$.
       -Otherwise, $S_g := \{i \in E_g \mid \beta_i^* = 1\}$.
       -Set the completion times of activities in $S_g$ $c_i := c_i^*$
•  end while
```

Decision variables of problem (8) are the chosen activities to be supplied by resources and the resource capacities assigned to those activities. The choice of the activities to be supplied with available resources at each decision point should also reduce the remaining projects duration as much as possible. Therefore, whilst the first term of the objective function tries to maximize the resource consumption at each decision point, the second term aims at reducing the partial makespan. Problem (8) has to be solved at each decision point, when at least more than one activity is ready to be operated and the residual available amount of resources is not zero.

3.3 Remark

Both the joint probabilistic constraints and the resource allocation heuristics involve the solution of a model under joint probabilistic constraints. In the following, we show how to transform them, in the case of independent random variables, into deterministic equivalent problems.

With this aim, let consider the following problem:

$$P\left(\hat{a}_i^T x \leq \hat{b}_i \; \forall i = 1,\ldots,m\right) \tag{9}$$

where the probabilistic constraints are jointly imposed on m separate constraints involving the random variables $_i$. Under the independence assumption among the random variables $_i$, the probabilistic constraints (9) can be rewritten as

$$\prod_{i=1,\dots,m} P(a_i^T\, x \leq$$

and equivalently, denoting with F_i the marginal probability distribution function of the continuous random variable b_i, as

$$\prod_{i=1,\dots,m} F_i\,(a$$

By taking logarithm we can rewrite the constraints as follows:

$$\sum_{i=1,\dots,m} \ln F_i\,(c_i^T\; x\,) \geq \ln \alpha.$$

Since the logarithm is an increasing function and $0 < F_i \leq 1$, this transformation is legitimate.

Furthermore, for log-concave distribution functions, convexity of the constraints is preserved. Fortunately, the class of log-concave random variables includes several commonly used continuous, univariate probability distributions as for example the Uniform, Normal, Exponential, Beta, Weibull, Gamma, Pareto, and Gompertz distributions. We observe that also in the case of discrete distributions, problems with joint probabilistic constraint can be reduced to deterministic equivalent problems that we shall report hereafter, for the sake of completeness. Let us introduce an integer vector z_i, whose entries are defined as $z_i = a_i^T\, x \quad i = 1\dots,m$. In the case of log-concave marginal distribution, it is possible to rewrite z_i in a 0 - 1 formulation. If $l_i + k_i$ is a known upper bound, where l_i represents the α-fractile of the distribution function F_i, z_i can be written as

$$z_i = l_i + \sum_{k=1\dots K} z_{ik}$$

and the probabilistic constraints as

$$\sum_{i=1,\dots,m} \sum_{k=1\dots k_i} a_{ik} z_{ik}$$

where

$$= \ln F_i(l_i + k) - \ln F_i(l_i + k - 1)$$

and

$$= \ln \quad - \ln F(l).$$

4. COMPUTATIONAL EXPERIMENTS

In this section we shall present the results of the computational study carried out with the aim of assessing the performances of the algorithms introduced in the previous section. We have compared our algorithm with the approaches found in the literature closer to our work (the STC and the RFDFF heuristics) and with a set of newly created benchmark heuristics, which rely on the use of separate chance constraints.

The computational experiments have been carried out on a set of benchmark problems with 30 and 60 activities randomly selected from the project scheduling problem library PSPLIB (Kolisch and Sprecher 1997).

We have replaced the deterministic duration of each activity by two probability distributions, one continuous and one discrete, taking d_i as their mean.

In particular, we have tested the Uniform distribution U($0.75d_i$; $2.85d_i$) and the Poisson distribution with mean d_i and activity durations are assumed to be independent.

Extensive simulation has been used to evaluate all procedures on robustness measures and computational efficiency. For every network instance, 1000 scenarios have been simulated by drawing different actual activity durations from the described distribution functions.

Using these simulated activity durations, the realized schedule is constructed by applying the following reactive procedure. An activity list is obtained by ordering the activities in increasing order of their starting times in the proactive schedule. Ties are broken by increasing activity number. Relying on this activity list, a parallel schedule generation scheme builds a schedule based on the actual activity durations. We opted for railway execution never starting activities earlier than their prescheduled start time in the baseline schedule. Effectively, this type of constraint is inherent to course scheduling, sports timetabling and railway and airline scheduling, or when activity execution cannot start before the necessary resources have been delivered.

4.1 Separate chance-constraints based heuristics

Schedule generation schemes are the core of most heuristic solution procedures for the deterministic RCPSP. The best-known Schedule generation scheme order all activities according to a priority list and at every decision point select the next activities to start based on this priority list. Mainly two types of schedule generation scheme are used to build a schedule from an activity list, the Parallel Schedule generation scheme and the serial schedule generation scheme. These schedule generation schemes construct a precedence-resource feasible sequence through a stepwise increase of a partial schedule. The serial method at each step, selects an activity from the set of activities eligible for scheduling and assigns to it the earliest possible start time according with both precedence and resource constraints. Selection of activities from the set of activities eligible for scheduling is performed according to a priority rule. The parallel method is based on a time incrementation procedure since at every decision time (t = 0 and the completion times of activities), it starts as many unscheduled activities as possible in accordance with the precedence and resource constraints selecting activities according to a priority rule. We use activity-based priority policies as benchmark heuristics, considering, instead of the deterministic durations, the α-fractiles of activities. We shall refer to these heuristics in the following as Parallel Separate chance-constraints based heuristics (PSCCBH) e Serial Separate chance-constraints based heuristics (SSCCBH).

Regardless the schedule generation scheme applied, the resulting schedule depends on the ordering criterion adopted. The following static priority rules for generating the priority list have been tested in the computational experiments (Kolisch and Hartmann 1999). Some of them have been proposed by the authors.

- The MinC rule orders the activities by increasing value of their resource requirement.
- The MinD rule orders the activities by increasing value of their α-fractile.
- The MaxC rule orders the activities by decreasing value of their resource requirement.
- The MaxD*C rule orders the activities Job ordered by decreasing value of their alfa-duration*resource requirement.
- The LST rule(Kolisch, Sprecher and Drexl 1995) orders the activities by increasing value of their latest starting time.
- The LFT (Davis and Patterson 1975) orders the activities by increasing value of their latest finish time.
- The MTS (Alvarez-Valdes and Tamarit 1989) orders the activities by decreasing value of the number of their successors.

5. ANALYSIS OF THE RESULTS

A total of 13 scheduling procedures are evaluated. Algorithms 1-4 are the PSCCBH whereas algorithms 5-8 are the SSCCBH with the first four priority rule listed in Section 4.1. Algorithms from 9 to 11 are the PSCCBH with the rules LST, LFT and MTS respectively, whereas Algorithm 12 is a SSCCBH with the LST priority rule. The JPCH Section 3.1 is executed considering the four priority rules (JPCH 1{4). All the algorithms, but the RFDFF and the STC heuristics, have been tested for 5 α values {0:8; 0:85; 0:9; 0:95; 0:99}.

The computational experiments were performed in a PC Pentium III, 667 MHz, 256 MB RAM. All procedures were coded in AIMMS language (Bisschop and Roelofs 2007) and the subproblems solved woth Cplex 10.1 and Conopt. We show in Tables 1-4 the average results calculated over all networks and executions for each approach. The complete set of the numerical results are fully reported in (Beraldi, Bruni, Guerriero and Pinto 2007). The quality of the algorithm on every problem instance has been evaluated by the following measures: average tardiness (Tavg) over all networks and executions, average timely project completion probability (TPCP) over all networks and executions, average number of jobs over all networks and executions, whose starting time in the actual schedule differs from the baseline schedule (delayed) and CPU time in seconds (time).

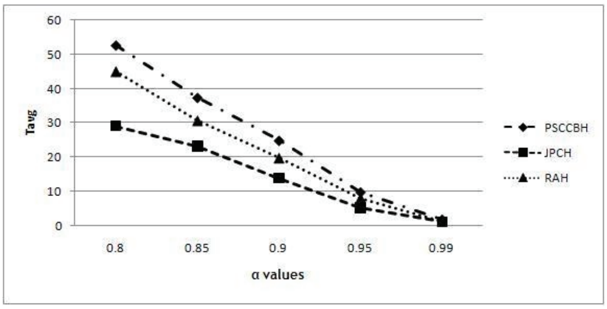


Figure 1: Tardiness versus α values

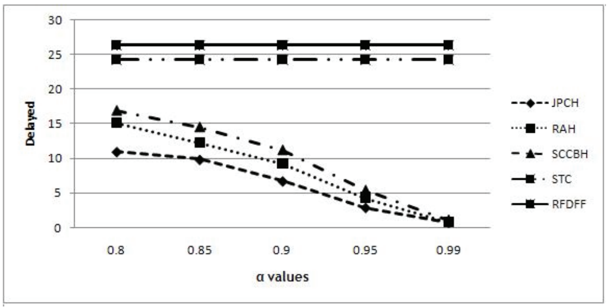


Figure 2: Delayed versus α values

For the sake of clarity, although RFDFF and STC heuristics construct exactly the same schedule whatever the risk averseness of the decision maker, we have reported the results of the RFDFF and STC heuristics for all the probability levels tested. As could be expected, proactive scheduling procedures, which use probabilistic information, always seem to outperform procedures that do not. It can be easily observed that even the simple benchmark heuristics 1-12 perform better than STC and RFDFF. In this respect, a note of caution is in order. The performances of the STC and RFDFF heuristics may have been biased by the high disruption probability associated with each activity. We have experimentally observed that, in the 1000 project simulations, almost all the activities showed a duration higher than their expected value, denoting a highly variable environment. This high variability is confirmed by the TPCP which is, as evident, on average very low.

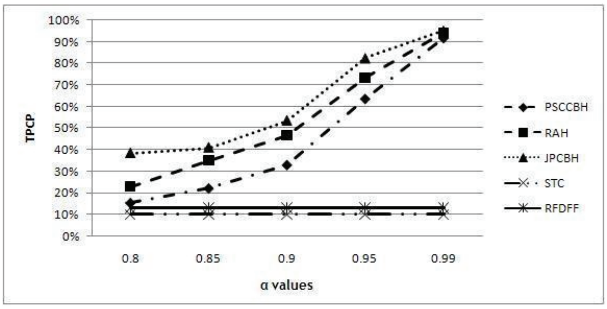


Figure 3: TPCP versus α values

JPCH 1-4 rank best among the heuristics. The gain in performances is more evident for increasing _ values, reflecting a more conservative strategy hedging against more disruptions scenarios. The performances of the algorithms 1-12 are clearly indistinguishable, and depend on the ordering criterion adopted.

Unfortunately, there is no unitary evidence of one criterion over the others. For these heuristics, the number of activities whose actual starting time exceed the planned starting time (reported in the column delayed) is quite satisfactory, especially for increasing α values. In particular, their performances are comparable with those of JPCH 1-4 for very high probability levels.

The RAH performs better than algorithms 1-12 but shows worse performances than JPCH 1-4.

With the aim of assessing the variation of the performance measures with respect to the probability level α we show in Figures 1, 2 and 3 the tardiness, the number of activities delayed and the timely project completion probability for different α values, for the same test problems. The average performance over the twelve separate chance constraints based heuristics has been considered for comparison. The tardiness of the STC and RFDFF heuristics has not been reported, since it was very high. As evident, the schedule performances deteriorate with decreasing α values. Similar behavior has been observed for all the other tests. This result is expected, since there is clearly a correlation between the schedule robustness and the values of α

Algorithms from 1 to 12 are computationally very cheap. This is due to the simple schedule construction procedures based on the parallel and serial schedule generation scheme. We observe that RFDFF and STC are slower due to the fact that the procedures have to be executed a number of times, until a creation of a deadline feasible schedule is allowed.

The computational time of JPCH 1-4 and of procedures STC and RFDFF is comparable for most instances.

However, we should observe that for larger networks JPCH 1-4 show worse computational times than STC and RFDFF heuristics. This is due to the extra effort required for solving, at each decision point, the probabilistic model. We notice, in addition, that depending on the nature of the probabilistic model to be solved (nonlinear continuous or linear integer), a different computational effort is required. In our experiments, it is evident that more CPU time is required to solve the integer deterministic equivalent problem related to discrete random variables. We observe also that for the continuous distribution function considered, the RAH is overwhelming from a computational point of view (no results are reported in Tables 3 and 4.

This is due to the mixed integer nonlinear nature of the problem to be solved at each iteration of the RAH.

As a final remark, we observe that our heuristic does not consider, in its present form, possible weights associated to activities representing the marginal cost of starting the activity earlier or later than planned in the baseline schedule. We remark, that the ordering criterion could be adapted to reflect the relative importance of the activities. Nevertheless, there is no absolute consensus on how to estimate such costs.

Table 1: Results on 30 nodes test problems with discrete duration variability

	alfa=0,8				alfa=0,85				alfa=0,9				alfa=0,95				alfa=0,99			
P	**Tavg**	**TcP**	**delayed**	**time**	**Tavg**	**TcP**	**delayed**	**time**	**Tavg**	**TcP**	**delayed**	**time**	**Tavg**	**TcP**	**delayed**	**time**	**Tavg**	**oMn**	**delayed**	**time**
1	52,12	0,17	17,09	0,05	37,89	0,20	14,83	0,06	26,55	0,30	11,82	0,04	9,78	0,68	5,40	0,05	1,69	0,93	1,10	0,05
2	53,04	0,15	16,74	0,05	36,47	0,20	14,34	0,06	22,49	0,36	10,64	0,05	8,98	0,63	5,17	0,05	1,66	0,91	1,12	0,06
3	58,04	0,14	17,53	0,05	38,73	0,19	14,64	0,07	28,11	0,30	12,21	0,05	10,07	0,62	5,61	0,06	1,73	0,91	1,14	0,06
4	51,37	0,15	16,90	0,05	37,89	0,23	14,83	0,06	24,38	0,35	11,26	0,05	8,80	0,68	5,04	0,07	1,69	0,92	1,16	0,06
5	53,38	0,16	16,79	0,23	37,19	0,25	14,38	0,24	25,82	0,36	11,15	0,22	10,74	0,62	5,81	0,24	1,65	0,94	1,07	0,24
6	57,19	0,15	17,30	0,23	37,85	0,23	14,36	0,26	25,80	0,33	11,25	0,21	10,96	0,59	5,79	0,26	1,90	0,91	1,19	0,28
7	50,39	0,17	16,60	0,26	37,13	0,25	14,02	0,29	25,20	0,35	11,06	0,23	9,09	0,67	4,82	0,27	1,87	0,91	1,19	0,27
8	53,63	0,17	16,88	0,17	38,42	0,23	14,52	0,27	25,10	0,37	11,01	0,21	10,00	0,64	5,34	0,25	1,72	0,92	1,12	0,25
9	52,10	0,14	17,13	0,05	35,51	0,24	14,47	0,04	23,38	0,30	11,09	0,04	9,80	0,59	5,82	0,04	1,64	0,91	1,11	0,08
10	50,62	0,15	16,99	0,05	36,45	0,22	14,61	0,05	23,78	0,33	11,21	0,04	9,56	0,65	5,35	0,04	1,52	0,94	1,03	0,09
11	50,37	0,14	16,97	0,05	38,16	0,18	15,34	0,04	24,18	0,28	11,31	0,05	9,30	0,62	5,49	0,05	1,61	0,92	1,11	0,12
12	50,20	0,14	16,64	0,31	36,58	0,22	14,43	0,31	22,70	0,31	11,03	0,31	9,21	0,63	5,50	0,30	1,54	0,91	1,08	0,40
13	45,05	0,23	15,16	0,41	30,70	0,35	12,32	0,35	19,75	0,47	9,29	0,35	8,06	0,74	4,34	0,35	1,22	0,94	0,79	0,35
JPCH-1	29,94	0,46	10,87	0,41	17,32	0,59	8,07	0,37	12,23	0,58	5,68	0,37	5,09	0,86	2,73	0,37	1,11	0,95	0,72	0,37
JPCH-2	32,33	0,28	11,95	0,41	28,58	0,32	11,70	0,37	17,80	0,41	8,78	0,37	5,45	0,81	3,04	0,37	1,23	0,95	0,90	0,37
JPCH-3	27,68	0,44	10,99	0,41	22,37	0,45	9,51	0,37	12,52	0,64	6,29	0,37	4,93	0,81	2,88	0,37	1,10	0,96	0,71	0,37
JPCH-4	26,30	0,35	9,97	0,41	24,25	0,27	10,01	0,37	12,87	0,52	6,14	0,37	5,31	0,82	2,95	0,37	1,00	0,95	0,68	0,37
STC	170,77	0,10	24,28	0,41	170,77	0,10	24,28	0,41	170,77	0,10	24,28	0,41	170,77	0,10	24,28	0,41	170,77	0,10	24,28	0,41
RDFF	73,78	0,13	26,39	0,41	73,78	0,13	26,39	0,41	73,78	0,13	26,39	0,41	73,78	0,13	26,39	0,41	73,78	0,13	26,39	0,41

Table 2: Results on 60 nodes test problems with discrete duration variability

	alfa=0,8				alfa=0,85				alfa=0,9				alfa=0,95				alfa=0,99			
P	**Tavg**	**TcP**	**jobrit**	**time**	**Tavg**	**TcP**	**jobrit**	**time**	**Tavg**	**TcP**	**jobrit**	**time**	**Tavg**	**TcP**	**jobrit**	**time**	**Tavg**	**TcP**	**jobrit**	**time**
1	170,64	0,01	41,3	0,17	126,35	0,02	41,4	0,16	73,51	0,12	33,0	0,15	33,38	0,43	19,0	0,156	5,80	0,85	4,1	0,16
2	154,63	0,01	39,2	0,17	114,36	0,03	39,6	0,17	73,05	0,14	31,9	0,15	29,63	0,45	16,5	0,174	5,76	0,87	3,9	0,16
3	165,49	0,01	40,0	0,17	127,90	0,03	41,2	0,18	79,37	0,12	33,4	0,17	35,35	0,38	19,6	0,161	5,92	0,84	4,1	0,16
4	156,20	0,01	39,7	0,18	120,15	0,04	40,6	0,18	71,62	0,15	31,6	0,17	31,39	0,43	17,4	0,156	5,35	0,87	3,7	0,17
5	150,72	0,02	37,4	0,76	107,25	0,09	36,6	0,75	71,33	0,18	29,7	0,78	31,10	0,46	16,6	0,723	5,24	0,88	3,4	0,74
6	155,62	0,02	38,1	0,61	114,95	0,07	38,4	0,62	69,17	0,20	29,3	0,59	30,46	0,48	15,6	0,575	5,06	0,88	3,3	0,59
7	148,88	0,03	37,7	0,82	111,81	0,08	37,5	0,82	68,06	0,15	29,5	0,80	29,77	0,49	16,0	0,736	5,50	0,87	3,5	0,75
8	141,47	0,04	37,1	0,61	107,49	0,08	36,6	0,64	67,81	0,20	29,0	0,63	28,03	0,51	15,0	0,623	5,63	0,87	3,7	0,62
9	161,14	0,01	44,4	0,12	120,34	0,02	40,8	0,11	77,18	0,08	34,0	0,10	32,02	0,47	18,5	0,103	6,33	0,84	4,4	0,10
10	160,62	0,01	44,6	0,12	118,85	0,02	41,0	0,10	76,95	0,08	34,1	0,12	31,87	0,41	18,1	0,108	6,11	0,83	4,3	0,11
11	160,01	0,00	44,4	0,14	119,13	0,03	40,9	0,11	72,05	0,09	32,8	0,12	30,96	0,35	18,3	0,111	5,46	0,86	4,0	0,10
12	151,12	0,01	43,5	0,74	107,12	0,04	38,5	0,66	72,42	0,12	32,3	0,72	28,81	0,51	16,5	0,641	5,34	0,86	3,8	0,67
13	115,80	0,10	37,0	2,29	74,61	0,31	30,3	2,29	41,74	0,51	20,8	2,29	24,41	0,63	13,5	2,286	3,93	0,90	2,7	2,29
20	32,64	0,40	10,9	2,40	26,79	0,47	8,9	2,40	22,18	0,62	6,9	2,40	31,25	0,84	8,8	2,403	30,20	0,97	7,9	2,40
21	33,96	0,41	11,8	2,40	37,08	0,46	12,1	2,40	38,65	0,62	11,9	2,40	41,02	0,81	11,2	2,403	29,52	0,96	7,7	2,40
22	27,02	0,40	9,9	2,40	32,45	0,57	10,9	2,40	25,80	0,63	7,4	2,40	28,82	0,80	8,4	2,403	26,53	0,97	7,4	2,40
23	22,86	0,38	9,0	2,40	25,37	0,56	9,3	2,40	21,56	0,68	7,1	2,40	21,50	0,82	6,8	2,403	26,72	0,97	7,0	2,40
STC	385,82	0,00	52,3	2,29	385,82	0,00	52,3	2,29	385,82	0,00	52,3	2,29	385,82	0,00	52,3	2,286	385,82	0,00	52,3	2,29
RDFF	193,29	0,03	58,5	2,40	193,29	0,03	58,5	2,40	193,29	0,03	58,5	2,40	193,29	0,03	58,5	2,403	193,29	0,03	58,5	2,40

Table 3: Results on 30 nodes test problems with continuous duration variability

	alfa=0,8				alfa=0,85				alfa=0,9				alfa=0,95				alfa=0,99			
P	**Tavg**	**TcP**	**jobrit**	**time**	**Tavg**	**TcP**	**jobrit**	**time**	**Tavg**	**TcP**	**jobrit**	**time**	**Tavg**	**TcP**	**jobrit**	**time**	**Tavg**	**TcP**	**jobrit**	**ex Norob**
1	27,99	0,27	13,04	0,04	14,07	0,44	9,27	0,04	7,11	0,59	6,24	0,06	2,59	0,84	2,48	0,04	0,00	1,00	0,00	0,04
2	27,74	0,26	12,81	0,05	13,54	0,45	9,02	0,05	7,57	0,62	6,45	0,05	2,65	0,84	2,50	0,04	0,00	1,00	0,00	0,04
3	32,14	0,23	14,26	0,09	17,12	0,36	10,98	0,05	8,20	0,56	7,00	0,06	2,51	0,85	2,37	0,05	0,00	1,00	0,00	0,04
4	26,57	0,28	12,36	0,04	13,99	0,41	9,39	0,06	6,81	0,60	5,93	0,05	2,34	0,85	2,22	0,05	0,00	1,00	0,00	0,05
5	30,39	0,25	13,55	0,21	14,96	0,41	9,70	0,21	7,18	0,60	6,18	0,21	2,40	0,84	2,28	0,19	0,00	1,00	0,00	0,19
6	30,42	0,25	13,63	0,18	13,96	0,44	9,39	0,20	8,05	0,58	6,86	0,20	2,45	0,84	2,34	0,17	0,00	1,00	0,00	0,18
7	29,93	0,25	13,70	0,19	15,60	0,38	10,09	0,23	8,52	0,52	7,18	0,20	2,18	0,86	2,11	0,19	0,00	1,00	0,00	0,20
8	28,82	0,25	12,90	0,17	14,28	0,41	9,42	0,19	8,06	0,56	6,82	0,19	2,38	0,86	2,26	0,17	0,00	1,00	0,00	0,18
9	27,32	0,24	12,98	0,04	14,58	0,38	9,56	0,05	7,34	0,59	6,35	0,05	2,22	0,88	2,13	0,04	0,00	1,00	0,00	0,05
10	26,22	0,29	12,74	0,05	14,20	0,42	9,53	0,05	7,56	0,58	6,52	0,05	2,30	0,88	2,19	0,04	0,00	1,00	0,00	0,05
11	27,59	0,26	13,20	0,05	14,57	0,38	9,68	0,04	7,39	0,60	6,40	0,05	2,40	0,82	2,30	0,04	0,00	1,00	0,00	0,05
12	26,63	0,26	12,88	0,25	13,65	0,40	9,11	0,26	7,36	0,64	6,37	0,26	2,25	0,86	2,16	0,25	0,00	1,00	0,00	0,26
20	12,09	0,85	5,83	0,15	3,26	0,90	2,33	0,16	1,66	0,94	1,42	0,15	0,55	0,97	0,47	0,16	0,00	1,00	0,00	0,15
21	26,92	0,69	12,04	0,37	14,70	0,74	8,36	0,37	7,30	0,85	5,35	0,38	1,77	0,94	1,53	0,37	0,00	1,00	0,00	0,36
22	22,31	0,71	9,78	0,23	9,90	0,83	6,47	0,23	5,77	0,88	4,47	0,23	1,58	0,95	1,20	0,24	0,00	1,00	0,00	0,24
23	28,00	0,64	11,64	0,20	15,57	0,77	8,74	0,22	6,41	0,87	4,82	0,19	1,73	0,93	1,42	0,20	264,41	1,00	0,00	0,22
STC	169,35	0,10	24,10	3,21	169,35	0,10	24,10	3,21	169,35	0,10	24,10	3,21	169,35	0,10	24,10	3,21	169,35	0,10	24,10	3,21
RDFF	82,62	0,12	27,28	3,58	82,62	0,12	27,28	3,58	82,62	0,12	27,28	3,58	82,62	0,12	27,28	3,58	82,62	0,12	27,28	3,58

Table 4: Results on 60 nodes test problems with continuous duration variability

	alfa=0,8				alfa=0,85				alfa=0,9				alfa=0,95				alfa=0,99			
P	**Tavg**	**TPCP**	**jobrit**	**CPU**	**Tavg**	**TPCP**	**jobrit**	**CPU**	**Tavg**	**TPCP**	**jobrit**	**CPU**	**Tavg**	**TPCP**	**jobrit**	**CPU**	**Tavg**	**TPCP**	**jobrit**	**CPU**
1	96,71	0,04	38,1	0,11	53,55	0,18	30,7	0,10	25,82	0,40	21,1	0,11	8,29	0,69	7,8	0,10	0,00	1,00	0,0	0,11
2	95,43	0,03	37,1	0,11	49,51	0,20	28,3	0,13	23,00	0,40	18,8	0,11	7,12	0,73	6,7	0,13	0,00	1,00	0,0	0,11
3	103,35	0,05	39,3	0,12	52,45	0,16	30,0	0,11	24,25	0,38	19,9	0,11	8,44	0,74	8,0	0,11	0,00	1,00	0,0	0,12
4	85,00	0,08	35,2	0,12	49,61	0,20	28,2	0,11	26,21	0,40	20,9	0,12	8,02	0,73	7,4	0,11	0,00	1,00	0,0	0,11
5	96,06	0,06	36,4	0,50	47,36	0,22	26,9	0,50	22,02	0,43	17,9	0,50	7,82	0,75	7,3	0,50	0,00	1,00	0,0	0,50
6	94,35	0,06	36,2	0,38	47,20	0,21	26,6	0,39	23,62	0,41	19,0	0,39	7,22	0,76	6,7	0,39	0,00	1,00	0,0	0,40
7	96,90	0,07	37,5	0,53	50,21	0,19	28,7	0,54	23,53	0,42	19,4	0,51	7,71	0,75	7,2	0,54	0,00	1,00	0,0	0,50
8	92,49	0,07	35,1	0,40	46,32	0,21	26,8	0,40	24,69	0,43	19,7	0,40	6,47	0,78	6,0	0,40	0,00	1,00	0,0	0,41
9	90,56	0,05	37,5	0,10	49,72	0,19	28,8	0,10	25,17	0,33	20,8	0,10	7,91	0,72	7,4	0,10	0,00	1,00	0,0	0,10
10	91,61	0,05	37,6	0,10	50,91	0,17	29,3	0,12	24,31	0,38	20,2	0,11	7,56	0,71	7,1	0,12	0,00	1,00	0,0	0,11
11	87,77	0,04	36,9	0,10	50,97	0,17	29,4	0,11	24,51	0,39	20,3	0,10	8,17	0,75	7,7	0,11	0,00	1,00	0,0	0,12
12	94,36	0,04	37,5	0,73	45,34	0,18	26,7	0,74	21,24	0,35	1233,6	0,74	7,93	0,68	7,4	0,74	0,00	1,00	0,0	0,75
A	5,53	0,96	3,1	0,35	3,77	0,92	2,3	0,51	0,81	0,99	0,8	0,84	0,38	1,00	0,4	0,51	0,00	1,00	0,0	0,86
B	27,67	0,95	14,2	1,56	14,18	0,96	9,0	1,57	6,54	0,98	5,6	1,95	1,90	0,99	1,8	0,51	0,00	1,00	0,0	0,86
C	50,76	0,85	16,3	1,05	13,51	0,96	9,4	1,30	6,56	0,98	5,9	1,95	1,49	0,99	1,5	1,57	0,00	1,00	0,0	1,96
D	50,52	0,86	15,1	1,02	16,15	0,95	10,7	1,30	5,66	0,97	5,3	1,09	2,35	0,99	2,3	1,30	0,00	1,00	0,0	1,58
STC	303,51	0,10	49,7	21,78	303,51	0,10	49,7	21,78	303,51	0,10	49,7	21,78	303,51	0,10	49,7	21,78	303,51	0,10	49,7	21,78
RFDFF	174,97	0,04	60,1	21,78	174,97	0,04	60,1	21,78	174,97	0,04	60,1	21,78	174,97	0,04	60,1	21,78	174,97	0,04	60,1	21,78

6. CONCLUSIONS

This paper presents heuristic procedures for solving project scheduling problems under uncertainty. The heuristics exploits probabilistic information on random activities duration within the framework of joint probabilistic constraints. In the proposed algorithm, the temporal aspect of the problem is treated at a higher level, whereas the probabilistic aspect is tackled at decision points, when activities are supplied by available resources. This hierarchical view of the problem has allowed to develop effective heuristics for projects with high variability with the aim of obtaining a schedule with good performances.

7. REFERENCES

Alvarez-Valdes, R., Tamarit, J.M., 1989. Heuristic algorithms for resource constrained project scheduling: A review and an empirical analysis. *Advances in project scheduling*, 113-134. Slowinski, R. and J. Weglarz (eds.). Elsevier, Amsterdam.

Artigues, C., Roubellat, F., 2000. A polynomial activity insertion algorithm in a multi-resource schedule with cumulative constraints and multiple modes. *European Journal of Operational Research*, 127, 297-316.

Artigues, C., Michelon, P., Reusser, S. (2003). Insertion techniques for static and dynamic resource-constrained project scheduling. *European Journal of Operational Research*, 149, 249-267.

Aytug , H., Lawley , M.A., McKay, K. , Mohan, S., and Uzsoy, R., 2005. Executing production schedules in the face of uncertainties: A review and some future directions. *European Journal of Operational Research*, 161, 86-110.

Ballestin, F., 2007. When it is worthwhile to work with the stochastic RCPSP?. *Journal of Scheduling*, 10, 153-166.

Beraldi, P., Bruni, M.E., Guerriero, F., Pinto, E., 2007 Heuristic procedures for the robust project scheduling problem, Technical Report, ParCoLab, DEIS, University of Calabria.

Bisschop, J., Roelofs, M., 2007. AIMMS. 3.7 Users guide. Paragon Decision Technology B.V., The Netherlands. Optimization, Inc., Incline Village, NV, 2006.

CPLEX, ILOG CPLEX 6.5: Users Manual, CPLEX Optimization, Inc., Incline Village, NV, 1999.

Drud, A. S. CONOPT: A System for Large Scale Nonlinear Optimization, Reference Manual for CONOPT Subroutine Library, 69p, ARKI Consulting and Development A/S, Bagsvaerd, Denmark (1996).

Davis, E., Patterson, J., 1975. A comparison of heuristic and optimum solutions in resource-constrained project scheduling. *Management Science*, 21, 944-955.

Deblaere, F., Demeulemeester, E., Herroelen, W., Van de Vonder, S., 2007. Robust Resource Allocation Decisions in Resource-Constrained Projects. *Decision Sciences*, 38, 1-37.

Golenko-Ginzburg, D., Gonik, A., 1998. A heuristic for network project scheduling with random activity durations depending on the resource allocation. *International Journal of Production Economics*, 55, 149-162.

Golenko-Ginzburg, D., Gonik, A., 1997. Stochastic network project scheduling with non-consumable limited resources. *International Journal of Production Economics*, 48, 29-37.

Herroelen, W., Leus, R., 2004. The construction of stable project baseline schedules. *European Journal of Operational Research*, 156, 550-565.

Herroelen, W., Leus, R., 2004. Robust and reactive project scheduling: a review and classification of procedures. *International Journal of production Research*, 42(8), 1599-1620.

Herroelen, W., Leus, R., 2005. Project scheduling under uncertainty - survey and research potentials. *European Journal of Operational Research*, 165(2), 289306.

Igelmund, G., Radermacher, F.J., 1983. Algorithmic approaches to preselective strategies for stochastic scheduling problems. *Networks* (13), 29-48.

Kolisch, R., Sprecher, A., 1997. PSPLIB a project scheduling problem library. *European Journal of Operational Research*, 96, 205-216.

Kolisch, R., Sprecher, A., Drexl, A., 1995. Characterization and generation of a general class of resource constrained project scheduling problems. *Management Science*, 41(10), 1693-1703.

Kolisch, R., Hartmann, S., 1999. Heuristic algorithms for solving the resource-constrained project scheduling problem: classification and computational analysis J. Weglarz (editor): Project scheduling: Recent models, algorithms and applications, 147-178, Kluwer, Amsterdam, the Netherlands, 1999.

Leus, R., 2003. The generation of stable project plans. Ph.D. Thesis, Department of Applied Economics, Katholieke Universiteit Leuven, Belgium.

Leus, R., Herroelen, W., 2004. Stability and resource allocation in project planning. *IIEE Transactions*, 36, 667-682.

Mehta, S.V., Uzsoy, R.M., 1998. Predictable scheduling of a job shop subject to breakdowns. *IEEE Transactions on Robotics and Automation*, 14(3), 365-378.

M hring, R.H. and Stork, F., 2000. Linear preselective policies for stochastic project scheduling. *Mathematical Methods of Operations Research*, 52, 501-515.

Rabbani, M., Fatemi Ghomi, S.M.T., Jolai, F., Lahiji, N.S., 2007. A new heuristic for resource-constrained project scheduling in stochastic networks using critical chain concept. *European Journal of Operational Research*, 176, 794-808.

Stork, F., 2000. Branch-and-bound algorithms for stochastic resource-constrained project scheduling, Research Report. 702/2000. Technische Universitat Berlin

Tavares, L.V., Ferreira, J.A.A., Coelho, J.S., 1998. On the optimal management of project risk. *European Journal of Operational Research* 107, 451-469.

Tsai, Y.W., Gemmil, D.D., 1998. Using tabu search to schedule activities of stochastic resource-constrained projects. *European Journal of Operational Research* 111, 129-141.

Van de Vonder, S., Demeulemeester, E., Herroelen, W., Leus, R., 2005. The use of buffers in project management: The trade-off between stability and makespan. *International Journal of Production Economics*, 97, 227-240.

Van De Vonder, S., Demeulemeester, E., Herroelen, W., Leus, R., 2006. The trade-off between stability and makespan in resource-constrained project scheduling, *International Journal of Production Research*, 44(2), 215-236.

Van de Vonder, S., Demeulemeester, E., Herroelen, W., 2007. A classi_cation of predictive-reactive project scheduling procedures. *Journal of Scheduling*, 10, 195-207.

Van de Vonder, S., Demeulemeester, E., Herroelen, W., 2008. Proactive heuristic procedures for robust project scheduling: An experimental analysis. *European Journal of Operational Research*, 189, 723-733.

Wang, J. (2005). Constraint-based schedule repair for product development projects with time-limited constraints. *International Journal of Production Economics*, 95, 399-414

Wiest, J.D., Levy, F. K., 1977 A Management Guide to PERT/CPM with GERT/PDM/DCPM and Other Networks, Prentice-Hall, New Jersey.

Zhu, G., Bard, J.F., Yu, G., 2005. Disruption management for resource-constrained project scheduling. *Journal of the Operational Research Society*, 56, 365-381.

Zhu, G., Bard, J.F., Yu, J.F., 2007. A two-stage stochastic programming approach for project planning with uncertain activity durations. *Journal of Scheduling* 10, 167-180.

HIGHWAY TRAFFIC MODEL BASED ON CELLULAR AUTOMATA: PRELIMINARY SIMULATION RESULTS WITH CONGESTION PRICING CONSIDERATIONS

S. Di Gregorio(a)**, R. Umeton**(b)**, A. Bicocchi**(c)**, A. Evangelisti**(d)**, M. Gonzalez**(e)

(a,b)Department of Mathematics, University of Calabria, Arcavacata, Rende, CS, 87036, Italy.
(c,d)Abstraqt srl, Lucca, LU, 55100, Italy.
(e)Department of Energy Technology, Royal Institute of Technology, Stockholm, SE-10044, Sweden.

(a)dig@unical.it (b)umeton@mat.unical.it (c)a.bicocchi@abstraqt.it (d)a.evangelisti@abstraqt.it (e)mags2@kth.se

ABSTRACT

Cellular Automata are a reputable formal support for traffic modelling and simulation. STRATUNA is a Cellular Automata model for simulating the evolution of two/three lane highways. It encodes the wide specification of driver's response to the events in his sight range. Encouraging comparison between simulated events and their corresponding in the reality bring to the specification of a theoretical general model characterized by an increased expression power and a significantly deeper forecasting potential, whose application fields are numerous and varied. Fair results in flow forecasting lead to the implementation of an established cost system in which simulation directly provides cost forecasting in terms of congestion toll.

Keywords: modelling, simulation, cellular automata, highway traffic

1. INTRODUCTION

In the field of modelling high complexity systems, Cellular Automata (CA) (von Neumann 1967) are a widely used computational paradigm to simulate those systems which evolve mostly according to an a-centrism schema through local interactions of their constituent parts. Basically, a CA can be seen as a *d*-dimensional space partitioned into uniform cells, each one embedding a computational tool, namely the elementary automaton (EA). If we consider this computational device as a black-box, the output of each cell is its internal state. On the other hand, input for these EAs is given by states of other neighbouring EAs. The neighbouring conditions are determined by a pattern invariant in time and in space. At the beginning, each EA is in an arbitrary state that defines the initial condition; afterwards, the CA evolves by changing concurrently states to all of the EAs at the same discrete time steps, according to EA transition function (parallelism property).

In the field of highway modelling, thanks to a-centric and parallel properties of such a system, CA are intensively used (Schadschneider 2006) for analysis of traffic emerging phenomena. In fact, when a highway exhibits fixed structural characteristics and there are no external interferences out of the vehicular interactions (normal conditions), the traffic evolution emerges by the mutual influences among next vehicles.

At our knowledge, the main CA models for highway traffic (Nagel and Schreckenberg 1992; Wolf 1999; Knospe, Santen, Schadschneider, and Schreckenberg 2000; Lárraga, Del Ríob and Alvarez-Icaza 2005) simulation can be considered "simple": external inputs to each EA and corresponding feedbacks are very regular and easy. However, these simple models reproduce the basic three different phases of traffic flow ("free flow", "wide moving jams" and "synchronized flow") and simulations are in general compared with single vehicle data (i.e. automatically collected data gathered by stationary inductive loops placed above highway lanes).

In this paper we present STRATUNA (Simulation of highway TRAffic TUNed-up by cellular Automata), a new model for highway traffic modelling based on CA. The first aim of our model is to express more accurately driver nearby conditions and reactions; a guide to easily interpret our model is that a vehicle and its driver are strongly coupled (and generally created according to a *normal* distribution or field data) and make a whole. We have based a $\beta 4$ implementation on an improvement of previous CA model (Di Gregorio and Festa 1981; Di Gregorio, Festa, Rongo, Spataro, Spezzano and Talia 1996), that was adequately rewarding in the past, but now it is out-of-date for the different industrial situations (for instance, the classification of vehicles on the base of mere acceleration/deceleration capabilities is now unrealistic). Field data used to feed this early model come from Italian highway A4 and are composed by timed highway entrance-exit specification coupled to vehicle type and approximated route length. These data make it possible the comparison between real event and simulated ones in terms of average speed.

The analysis of problems that come out from the adoption of the implemented model, together with a deep theoretical study, lead to the general STRATUNA model, whose specification is one of the objects of this paper and promises profound forecasting capabilities.

Finally, a cost system is proposed in which the simulation model finds application and fits well in cost forecasting. Conclusions are reported at the end of the paper.

2. STRATUNA $\beta 4$

STRATUNA $\beta 4$ is, at present, a representative implementation and is based on an extend CA definition, namely "macroscopic" (Di Gregorio and Serra 1999). The extension in fact introduces "substates" and "external influences"; a substate defines an attribute of a cell and can be either self-sufficient or composed by other sub-substates and so on, as attributes in reality. On the other hand, the only external influence is represented by the entrance of vehicles at tollgates, measured out according to field data or probabilistic function.

We use a one-dimensional CA (since the y-position of a vehicle in a cell can be stored as substate) where each cell corresponds to 5m of highway in the reality. Inside this one-dimensional CA, in order to manage highway zones with respect to drivers, we introduce the notion of "free zone": each vehicle controls a disjunction of zones that are free of vehicles and cannot be reached by surrounding vehicles in the next step. To refine this concurrency control set up with "free zones", the indicator light is introduced, by means of which vehicles reveal their intentions planned for the next step. Moreover, as just anticipated, we introduce actions that last more than one step and optionally provide a kind of synchronization between drivers (e.g. the overtake action is done in more consecutive steps).

In order to complete given intuitions, we formally state now the implemented model:

$$STRATUNA_{\beta 4} = \left\langle R, E, X', P', S', \gamma_{\beta 4}, \tau_{\beta 4} \right\rangle \qquad (1)$$

$R = \{x \mid x \in N,\ 1 \le x \le n\}$ is the set of *n* cells, forming the highway, where *N* is set of natural numbers.
$E \subset R$ is the set of entrance-exit, special cells in *R* where vehicles can be introduced or eliminated.
$X' = \left\langle -r,\ -r+1,\ \ldots,\ 0,\ 1,\ \ldots,\ r \right\rangle$ defines the EA neighbouring, where *r* is a radius defining the sight range of the average driver, with $\#X = 2r+1$.
$P' = \{length,\ width,\ clock,\ lanes\}$ defines global parameters: *length* and *width* define the geometry of the cell, over x and y axis respectively; *clock* is the CA clock, *lanes* is the number of highway lanes (numbered 1, 2 .. from right to left) including additional lane 0, representing, from time to time, the entrance or exit or emergency lane.
$S' = \left\{ Static' \times \left(Vehicle' \times Driver' \right)^{lanes} \right\}$ represents the structure of substates and sub-substates that characterize every cell state. To be more precise, is proposed this hierarchy of substates in Table 1.

CellNO is a kind of ID that uniquely identifies a cell in the whole CA. *SpeedLimit* is the speed limit imposed and is specified lane by lane. After these static characteristics of a cell, we have the characteristics of a vehicle: *Type* is one in {motorcycle, car, bus/lorries/vans, semi-trailers/articulated}, while *Length* is the vehicle extension over the x-axis direction. Then we have the substates that code characteristics related to the speed and the acceleration of a vehicle. *Indicator'* can assume a value in {0,-1,1} respectively showing that {no, right, left} indicator is turned on. Finally, *Xposition* and *Yposition* collocate the middle point of the front side of the vehicle inside the cell. Going forward, we define driver's substates: the trip information (*Origin*, *Destination* $\in E \times E$), the desired speed and the suitability of every lane from driver's point of view.

```
BEGIN: TransitionFunction()
        FindNeighbours(); ComputeSpeedLimits();
        ComputeTargetSpeed(); DefineFreeZones();
        AssignLowSuitabilityWhereAFreeZoneIsCutted();
        if(ManouvreInProgress())
          continueTheManouvre(); return;
        if(myLane==0) //I'm on a ramp
          if(IWantToGetIn())
            if(TheRampEnded())
              if(ICanEnter()) enter(); return;
              else if(IHaveSpaceProblemsForward())
                  slowDown(); return;
                else followTheQueue(); return;
            else //the ramp is not ended yet
              if(IHaveSpaceProblemsForward())
                followTheQueue(); return;
              else keepConstantSpeed(); return;
          else //I want to get out
            if(TheRampEnded()) deleteVehicle(); return;
            else if(IHaveSpaceProblemsForward())
                followTheQueue(); return;
              else keepConstantSpeed(); return;
           //end lane==0
        else if(myLane==1)
          if(MyDestinationIsNear())  slowDown();
          if(MyDestinationIsHere())  goInLowerLane();
        else //myLane==2 or more
          if(ICanGoInLowerLane())
            if(GoingInLowerLaneIsForcedOrConvenient())
               goInLowerLane();
          else //I cannot go in lower lane
            if(MyDestinationIsNear())
              slowDown(); goInLowerLane();
        if(!IHaveSpaceProblemsForward()) //every lane
          if(TakeoverIsPossibleAndMyDestinationIsFar())
            if(TakeOverIsDesired()) takeover();
            else followTheQueue();
          else followTheQueue();
        else //I have space problems forward
          if(TheTakeoverIsForced()) takeover();
        return;
END;
```

Table 1: How Sub-substates Compose Substates

Substate	Composing sub-substates
Static'	*CellNO*, $SpeedLimit^{lanes}$
Vehicle'	*Type, Length, MaxSpeed, MaxAcceleration, MaxDeceleration; CurrentSpeed, Indicator', Xposition, Yposition,*
Driver'	*Origin, Destination, DesiredSpeed,* $Suitability^{lanes}$

$\gamma_{\beta 4}: N \times E \rightarrow Vehicle \times Driver$ is the *normal* generation function that introduces new pairs <*Vehicle, Driver*> into the highway at discrete steps.

$\tau_{\beta 4}: S'^{\#X'} \rightarrow S'$ is the EA transition function. Since this function is the core of the evolving model, a pseudo-code block is proposed before Table 1 with the aim of stating it; in that context, (1) "return" ends the evolution of single EA at each evolution step; (2) functions starting in lowercase are actions en-queued to be performed in further steps; (3) underlined functions represent the beginning of a synchronized protocol actuated over consecutive steps (e.g. actions in consecutive steps of takeover-protocol are: control a "free zone" on the left, light on the left indicator, start changing *Yposition*, and so on).

2.1. Field data composition and treatment

A clear and unambiguous way to understand how our model is fed, is to describe data used to define what we consider "reality" and what we consider a "real event".

Entrance-destination matrices referring to 5 non-contiguous weeks are evaluated in order to feed the generation function $\gamma_{\beta 4}$. Data are given by about 1 million digitalized toll tickets, pertain to Italian highway A4 Venezia-est/Lisert. This highway is characterized by twelve fundamental entrances/exits, two base lanes, and is long approximately 120km. Toll ticket data are, contextually with emission, grouped in five categories depending on number of axles and vehicle height at first axle (Cfr. Table 2; however, this classification is reducible to our vehicle classification as defined in substate *Type*). Due to some inconveniences of synchronization between tollgates, these datasets require a data cleaning step that involves the following groups of tickets: (i) missed tickets: transits without entrance or starting time; (ii) transits across two or more days; (iii) trips that end before they begin; (iv) vehicles too fast to be true (exceeding 200 km/h as average speed). In the aggregate, the cleaning interested about 10% of total tickets.

Subsequently, each one of the 34 days, is analyzed with respect to the average speed and the total flow; these measurements are compared to averages over all days. The result of this quantitative study is fixed in Figure 1: each dot represents one of the 34 analyzed days: a shift over x-axis and y-axis is a variation respectively of "total flow" and "average speed" from the value of the mean day. *DesiredSpeed* distributions for each vehicle *Type*, as presented in Table 2, are easily deduced by highway data for vehicles covering short distance in optimal conditions of flow and weather; since trips taken into account are really short, the probability of a rest or a need of refuel is minimal. However, in general data scenarios, vehicles that parked for a while in the rest and service areas cannot be detected by data and introduce errors; these kinds of errors justify the slightly higher values of average speed, obtained in the simulated cases, when compared with values corresponding to real events.

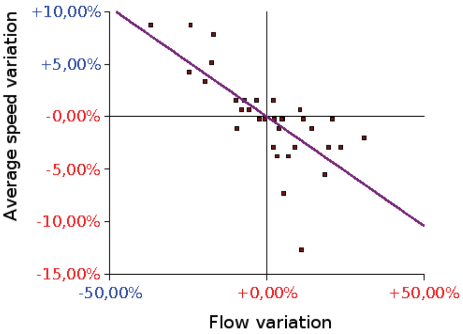


Figure 1: Daily Flow and Speed Fluctuation from the Average

Table 2: Share and Desired Speed for each Type of Vehicle in Selected Case of Free Flow

Vehicle type	Desired speed	Flow share
I	122,80 km/h	93,4%
II	112,77 km/h	4,6%
III	113,29 km/h	0,5%
IV	102,61 km/h	0,1%
V	93,90 km/h	1,4%

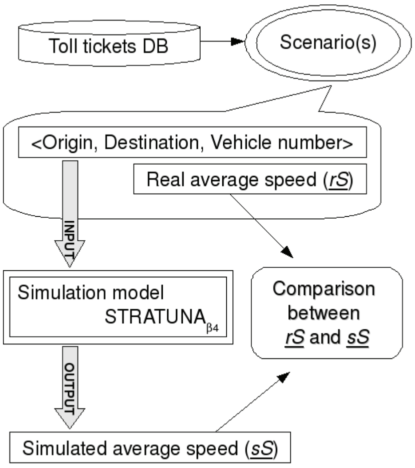

Toll tickets DB
Scenario(s)
<Origin, Destination, Vehicle number>
Real average speed (rS)
INPUT
Simulation model STRATUNA β4
Comparison between rS and sS
OUTPUT
Simulated average speed (sS)

Figure 2: Base Workflow, from Field Data to Simulations and Comparison with Reality

Finally, a statistical sampling treatment is performed to select meaningful subsets and three groups of possible scenarios are singled out: free flow situations, moderated flow cases, and cases in which the flow is higher than moderated and concentrated in a small region of the highway. These scenarios are what we refer to as "real events" and are used as touchstone in simulations presented in the following. To be more precise, each scenario provides a number of vehicles (each one specified by the couple <Origin,Destination>) and the average real speed (*rS*) over all its vehicles and over all the event, as shown in Figure 2.

2.2. Simulation results and comparison with reality

Despite the capabilities of model $\beta 4$, which can take into account different types of vehicle, generated vehicles are all cars, which represent, in fact, *95%* of real traffic. Starting from typical highway conditions, we report five representative simulations: one for free flow (Figure 3), two for moderated-flow next to congestion (Figure 4 and Figure 5) and two for locally congested situations (Figure 6). The following information are presented in each figure: corresponding real average speed *rS* (painted as a line), step-by-step average simulated speed *sS* (represented as a fluctuating curve), and the desired speed *sDS* (represented as an invariant notch compiled with data presented in Table 2).

At the beginning of each simulation, the modelled highway is empty; then, it is fed, entrance-by-entrance, with vehicles according to the generation function $\gamma_{\beta 4}$. Because of this, and taking into account the fact that each generated vehicle starts from null speed, the initial *sS* value is very low. After this, we have a pump transient where a portion of the intake process takes place; during this transient, *sS* grows pointing the *sDS* value since each vehicle can move forward its desired speed. An increasing number of vehicles reverses this trend and adjusts the *sS* value according to emerged situations. When this "pump phase" ends (this is after about 500 simulated seconds), an "evolving phase" begins: the *sS* measure starts evolving according to emerged events, triggered by transition function through local interaction, synchronization events and driver behaviours induced by environment. This evolving phase and the whole simulated event are compared with real event by means of two error measurements, respectively e_1 and e_2. The e_1 value is calculated as the average relative error (over all CA steps) between *sS* and *rS*, while e_2 is the same as e_1 but calculated excluding the "pump phase". Now we present simulation results of the free flow scenario.

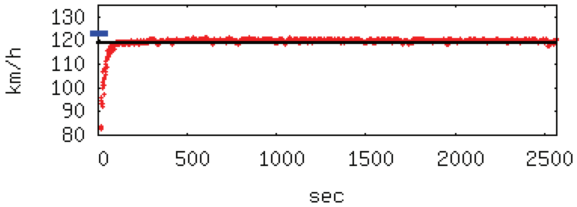


Figure 3: The Free Flow Case

In the free flow scenario presented above, the average simulated speed (*sS*) matches its equivalent in the reality (*rS*) during the whole simulation, outstanding slightly higher than field data, with very short oscillations (Figure 3). Indeed, we have recorded $e_1=1{,}29\%$ and $e_2=0{,}86\%$. Then we simulate another scenario, previously referred to as the moderated flow, in which the increased number of vehicles proposes circumstances next to congested situations.

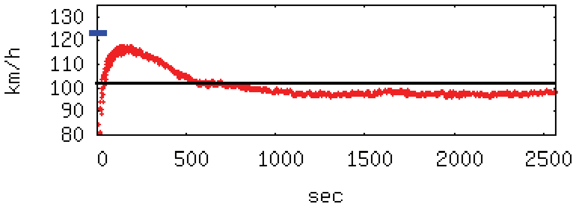


Figure 4: The Moderated Flow Case

In this scenario (Figure 4), after the same initial phase, *sS* becomes significantly lower than *rS* with moderate oscillations. Such a behaviour is unrealistic, despite of its low error rate: the cars in the simulation have to be faster than corresponding real cars, because they are "simpler" (they neither stop to give driver a rest neither need to refuel). This "slow moving" clearly depends on the fact that the simulated driver makes his subjective evaluation in a too much cautiously way. This cautious evaluation comes out because of the partial implementation of the transition function, which reduced the moving potentiality (reaction rigidity). Since a solution could be obtained by setting a shorter time step, that is equivalent to a shorter reaction time, only for next simulation, the value of parameter *clock* is changed from 1s (Figure 4) to 0,75s (Figure 5), without altering any other input to the simulation. The result of this tuning in the reactivity of drivers leads to a more realistic simulation, with error values $e_1=6{,}47\%$ and $e_2=5{,}83\%$.

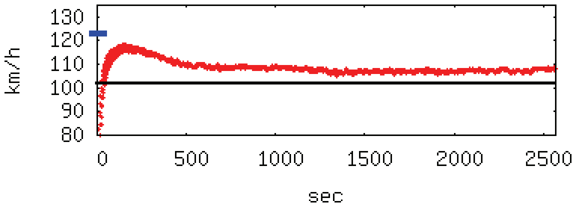


Figure 5: The Moderated Flow Case, with Tuned Reactivity Value

We present now two more simulations, with an implementation performance that is lower than the previous ones (Figure 6). Both simulations (lighter and darker oscillating curve) consider the particular, real, situation when a huge vehicle flow occurs only from one entrance and a climate issue alters driver behaviour (data refer to a really hot day in July concomitant with a migration from the city to the sea area). Both cases run on the same specifications of previous simulations, but *sS* becomes quickly significantly higher than *rS*. This means that, in this particular case, the reaction rigidity of the driver is rewarded by a higher speed due to a synchronization created by the forced filtering at the entrance. Corresponding error rates for lighter simulation are: $e_1=14,79\%$, $e_2=13,94\%$, while for the darker simulation we have $e_1=17,27\%$ and $e_2=17,12\%$.

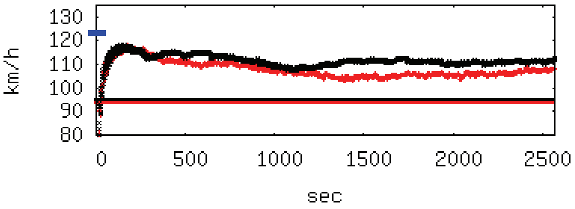


Figure 6: Locally Congested Situation

A chart summarizing the goodness of the model in speed forecasting is presented in Figure 7. Classical patterns of highway traffic (i.e. wide moving jams and synchronized flow) have been observed in simulations fed with synthetic (high) flow data. However the lack of single vehicle data (or data collected automatically by stationary inductive loops, as cited and presented by Schadschneider, 2006) does not enable a serious comparison between simulated patterns and real ones.

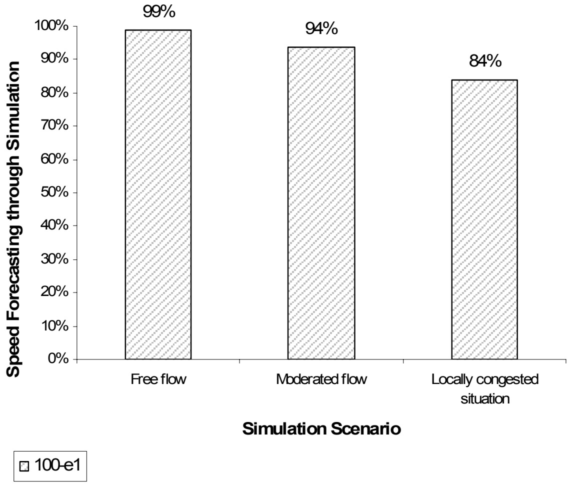


Figure 7: Goodness of Model in Speed Forecasting Task on Considered Scenarios

3. THE STRATUNA GENERAL MODEL

Encouraging results obtained with model $\beta 4$, along with the accurate analysis of its weakness, lead to the design of the extended mathematical model that is presented in this section.

Assuming that all the theoretical and practical structure previously detailed remain valid, we can go directly to the formal definition of the STRATUNA general model, stated by the following 8-tuple:

$$STRATUNA = \langle R,E,X,P,S,\mu,\gamma,\tau \rangle \qquad (2)$$

Here, besides R and E, some components are extended to provide more expressiveness of the model in order to tackle rigidity limitations discussed in the previous section.

$X = \langle -b, -b+1, \ldots, 0, 1, \ldots, f \rangle$ defines the neighbouring pattern of every EA, limited by furthest cells, forward (*f*) and backward (*b*), according to average driver's sight when visibility is optimal (no fog, no clouds, sunlight, etc.).

$P = \{ length, width, clock, lanes, \overline{weights} \}$ is the same set of global parameters as presented in (1), with the addition of a vector of weights that is responsible for outlining the behaviour of the average driver in different situations. This vector will be detailed with the transition function of the model.

$S = \{ Static \times Dynamic \times$
$\times (Vehicle \times Driver)^{lanes} \}$

specifies the high level EA substates, grouped by typologies. To be more precise, the first and the second group of substates model the statical and the dynamical features of the highway sector where the cell is located, respectively. Then, we have substates that model couples vehicle-driver, which are present in the quantity of at most one couple for each lane in the same cell. Each element of the set *S* is composed by sub-substates, as detailed in Table 3.

Table 3: How Sub-substates Compose Substates

Substate	Sub-substates self-explanatory names
Static	*CellNO, Slope, CurvatureRadius, SurfaceType, SpeedLimit, Lane1SpeedLimit, Lane0SpeedLimit*
Dynamic	*BackwardVisibility, ForwardVisibility, Temperature, SurfaceWetness, WindDirection, WindSpeed*
Vehicle	*Type, Length, MaxSpeed, MaxAcceleration, MaxDeceleration; CurrentSpeed, CurrentAcceleration, Xposition, Yposition, Indicator, StopLights, WarningSignal*
Driver	*Origin, Destination, DesiredSpeed, PerceptionLevel, Reactivity, Aggressiveness*

In addition to already detailed sub-substated and self-explanatory ones, we consider also *PerceptionLevel*, *Reactivity* and *Aggressiveness*, but we postpone argument of details where transition function is discussed. Evolution of the model is demanded to the external influences functions μ and γ, while interactions among vehicles are delegated to the transition function τ.

$\mu : N \times R \rightarrow Dynamic$ is the "weather evolution function", that can change values of substates in *Dynamic* substates if appropriated., for each step $s \in N$ and for each cell $c \in R$,

$\gamma : N \times E \rightarrow Vehicle \times Driver$ is the generation function that places pairs vehicle-driver, at certain steps, in cell in *E* (corresponding to tollgates). This generation function has been preferred to uniform intake rates in order to serve both field-data-driven input and *normal* distribution.

$\tau : S^{b+1+f} \rightarrow S$ is the EA transition function. It is important to notice that *b* and *f* are generally restricted in *b'* cells backward and to *f'* cells forward, according to visibility: cells out of driver's sight are considered without information.

The transition function τ, which represents the core of model evolution, is now widely detailed and exhaustively treated and can reveal the principal design choices concerning STRATUNA.

3.1. General model transition function

While highway characteristics are already partitioned in *Static* and *Dynamic* substates, both *Vehicle* and *Driver* contain constant and variable substates. It is important to separate statical and dynamical parts in the couple vehicle-driver because the main mechanism of traffic evolution is related to the assignment of new values to variable substates of this couple. What can change at each step in variable substates of *Vehicle* is: *Xposition* (individuating x coordinate of the middle point of vehicle front side, inside the cell), *Yposition* (same point as just cited, but in orthogonal axis and where a fraction value corresponds to occupy two lanes), *CurrentSpeed*, *CurrentAcceleration*, *Indicator* (which can assume one value from: null, left, right, hazard lights), *StopLights* (that can be only turned on or off according to a breaking process) and *WarningSignal* (on or off depending on the attention a driver needs). Moreover, we have driver's dynamical substates: *DesiredSpeed*, *PerceptionLevel*, *Reactivity* and *Aggressiveness*. A case in which a driver changes his desired speed is the accident case: he turns on hazards lights and his desired speed becomes null, without being immediately removed from highway. Other sub-substates in *Driver* or *Vehicle* are considered constant characteristics (e.g. the sub-substate *Type* models the macro category a vehicle belongs to: motorcycle, car, bus / lorries / vans, semi-trailers / articulated). By changing the value of *Indicator* or *StopLights* or *WarningSignal*, a vehicle can communicate its intentions to next vehicles; then a communication protocol starts, which induces EA to "ask each others" before performing actions, in order to avoid dangerous circumstances or deadlocking situations (e.g. an overtake starts from lane 1 to lane 2 concomitant with the end of another overtake from lane 3 to lane 2). *WarningSignal* is activated when a driver needs the lane portion immediately ahead of his vehicle to be free; the mechanism provides that the driver who perceives this warning adjusts temporary his behaviour, to give way. The behaviour is the key of every driver: sub-substates *PerceptionLevel*, *Reactivity*, *Aggressiveness* respectively describe a how much a driver pays attention to others, how long does he take to react to a stimulus and how emphasized his reaction will be.

Each moving vehicle *V* performs a set of operations and evaluations with a precise order; performed computations can be grouped from driver's point of view as "reading environment state" and "decide what to do". The former process is divided in objective and subjective perceptions. Objective perceptions are basically temporary changes of *Static* substates via *Dynamic* inputs, causing alterations of *Vehicle* characteristic. For instance, highway *surface_slipperiness* is derived by SurfaceType, *SurfaceWetness* and *Temperature*, then, this data are related to the *Vehicle* sub-substates in order to calculate the temporary variable *maximum_safe_speed* that accounts for the vehicle stability, speed reduction by limited visibility and speed limits in the lane, occupied by the vehicle; then *desired_speed* is set as minimum between *max_speed* and *DesiredSpeed*. Likewise, *Slope* and *surface_slipperiness* determine the temporary variables *max_acceleration* and *max_deceleration*, correction to sub-substates *MaxAcceleration*, *MaxDeceleration.* Proceedings in objective perception, the next step consist in the identification of "free zones" for *V*, i.e. all the zones in the different lanes, that cannot be occupied by the vehicles next to *V*, considering the range of the speed potential variations and the lane change possibility, that is always anticipated by *Indicator*; note that the possible deceleration is computed on the value of *max_deceleration* in the case of active *StopLights*, otherwise a smaller value is considered.

The next computation stage involves the subjectivity of the driver and consist in the behaviour for next step. First of all, actual *CellNO* is compared with the cell number reported in *Destination* in order to evaluate if the exit is close enough to force a really calm way of driving, opportunely waiting the exit ramp. If the destination is far, driver's aspire is to reach and maintain his *desired_speed*. To do this, in general several alternatives are available; to measure goodness of each option, a cost is linked to it, according to standard cost functions fixed by the vector *weight* in parameter set. Among all the possible options, the driver chooses the one with minimal sum of the costs.

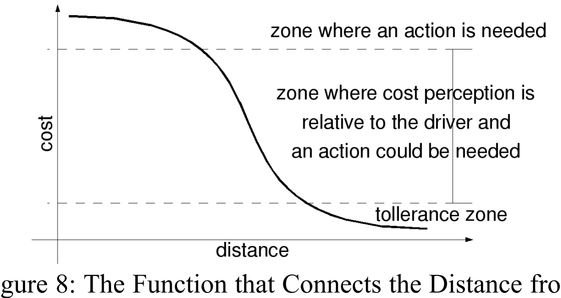


Figure 8: The Function that Connects the Distance from Front Vehicle with a Cost

An example of subjective process is now proposed. In Figure 8, a particular cost curve is presented, as individuated by *weight*, that connects the distance from front the vehicle with a penalty; in particular, this curve evaluates the gap between a vehicle and the one in front of it, which is supposed to suddenly break from time to time. Thus, the final choice is based on a driver subjective perception and evaluation of an objective situation by sub-substates *PerceptionLevel*, *Reactivity*, *Aggressiveness*.

Other examples of actions that lead to a cost are “remaining in a takeover lane”, “perceive a warning signal”, “staying far from *desired_speed*”, “breaking significantly”, “starting a takeover protocol”. P*erceptionLevel* concerns the perception of the free zones, whose length is reduced or (a little bit) increased by a percentage; however this “customized perceptions” are done in “customized security conditions”, considering the variable part of *Vehicle* substates, adding *max_speed*, *max_acceleration* and *max_deceleration*. *Aggressiveness* forces the deadlock that could rise during computation. For instance, when the entrance manoeuvre is prohibited due to a congested highway, since free zones are strongly reduced, the immobilised condition leads to a cost that increases significantly at each step; the variation of *Aggressiveness* value (a driver that is stressed by a cost) implies a proportional increase of the percentage value of *PerceptionLevel* until the free zone in a lane remains shorter than the distance between two consecutive vehicles. This leads to drivers that are temporary more aggressive, and can, therefore, perform the manoeuvre; when the manoeuvre is ended, and the generating cost too, *Aggressiveness* comes back to its original value.

As a result of illustrated features, the present general model has the needed expressivity to resolve problems that came out with model $\beta 4$. Indeed, the problem that the speed in the model is lower than the corresponding ones in the reality (Cfr. simulation in Figure 4) can be faced by *normally*-tuning the *Reactivity* value of simulated drivers, without altering the time step (*clock* parameter). Moreover, the possibility of reproducing locally congested situations got worse when climatic issues is introduced with the “weather function” μ. A serious problem that can rise with the adoption of sub-substates that describe average driver subjectivity, is the sure lack of field data. Without these data, that assign values of *PerceptionLevel* × *Reactivity* × *Aggressiveness*, the general model cannot lead to simulations. To overcome this obstacle, in the future we will employ the model inside a Genetic Algorithms (GA) (Holland 1975) to calibrate all included thresholds. The idea is that from the analysis of synthetic data and their comparison with the reality, we can infer those field data that characterize the real event, and use these “validated” deduced data to feed subsequent simulations (D'Ambrosio, Spataro and Iovine 2006).

3.2. Possible applications of the general model

Once the general model has been validated, a traffic forecasting analysis can be started, considering aspects that go beyond a simple comparison between simulated and real average speed. The expressivity taken into account from the general model suggests applications of the model in several fields related to traffic problems.

A promising use of STRATUNA model, probably through a GA, can be the study related to how highway owner should perform maintenance tasks in order to minimize road condition alterations during services. This can be summarized as “yard impact on road availability” and can be done at different levels: a starting level could be the yard planning such that the average simulated speed doesn’t decrease too much. A second level can be the design of better (that means safer or more reliable, etc..) highways through the simulation and the evaluation of different build options. Another field where the present model finds application is the evaluation of right price of a toll ticket, according to vehicular flow and private cost of each car. The basic idea is that a car which uses the highway reduces the time-gain derived from highway usage for other motorists; moreover, this travelling vehicle reduces the quality of life (by introducing pollution, noise, etc.) even of non-travelling people; Road pricing strategy searching is a widely diffused problem: a congestion toll system let motorists recognize the total cost they are imposing to other and has the good side effect of reducing vehicular flows. Since the STRATUNA general model exhibits forecasting features that can provide input information for a particular cost system, we now propose a preliminary cost study where our model can lead to a cost forecast.

4. PRELIMINARY STRATUNA-DRIVEN COST SYSTEM FOR CONGESTION TOLL

Theories on congestion pricing have been under research since the 1920’s and there are numerous references in literature about methods to estimate the costs for operating a car (fuel costs, maintenance, etc) in addition to the costs that each individual traveller imposes on other travellers due to the fact that each car increases the congestion of the highway. Road pricing has been implemented in various countries worldwide in order to reduce the traffic congestion problems in urban roads and highways. Here we propose an established cost system in which the simulation model can guide to quality of life and business advantages.

Assuming all vehicles are only cars, the principle of congestion pricing (Pigou 1920) provides a direct curve of correlation between traffic volume and its costs. In fact, every motorist making a trip introduces personal expenses in terms of private marginal costs, *MC*, (that are operating car costs plus the value of time spent in the highway) and takes a social cost (whose average will be denoted as *AC*). The difference between *MC* and *AC* represents the cost that a driver induced on his road neighbours (Li 2002): if *c* is the hourly average generalized travel cost (as above, it is composed by car operating costs plus value of travel time) and is supposed to be invariable, *dist* is the covered distance (assumed to be 1 km in the second part of Eq. 3), *V(q)* is a function of the flow *q* and represents the speed of vehicles, then *AC*, with respect to a certain flow value *q*, is given by:

$$AC(q) = c\frac{dist}{V(q)} = \frac{c}{V(q)} \qquad (3)$$

Thus the total cost *T(q)* of those vehicles is simply *T(q) = q AC(q) = (qc)/V(q)*. This means that for each new vehicle joining the flow *q*, we have the following marginal cost for the community:

$$MC(q) = \frac{d}{dq}\left(T(q)\right) =$$

$$= \frac{V(q)c - qc\frac{d}{dq}(V(q))}{V(q)^2} = \qquad (4)$$

$$= AC(q) - \frac{qc}{V(q)^2}\cdot\frac{d}{dq}(V(q))$$

Assuming that MC increases much more rapidly than AC when congestion begins (i.e. a flow *q > q'*), the difference between these two values is the considered money that motorists have to pay if we want to charge the cost they are imposing to the society. This means that the "congestion toll" *r* is given by:

$$r = MC(q') - AC(q') =$$

$$= -\frac{qc}{V(q)^2}\cdot\frac{d}{dq}(V(q)) \qquad (5)$$

This quantity could be equal to zero when there is no congestion (i.e. flow *q <= q'. Cfr. Fig. 9*), increases when the flow increases and subsequently decreases when *V(q)* increases. Now we introduce a model that is widely used and empirically verified over several highway models to establish the correlation between the flow and the speed of vehicles composing it: the Drake model (Drake 1967). Let q_0 be the maximum flow capacity (vehicles per hour per lane), V_0 the corresponding speed at maximum flow capacity and V_f the speed in free flow condition, then in the framework of Drake model, *q* is given by:

$$q = V(q)\cdot\frac{q_0}{V_0}\cdot\sqrt[\delta]{\delta\cdot\ln(V_f / V(q))} \qquad (6)$$

The speed-flow relationship given by Eq. 6 where δ is a parameter equal to *2* (Drake 1967), can be used inside Eqs. 3-5 to estimate the congestion toll when the flow is higher than *q'* (*Cfr. Fig. 9*) and the Drake model is an accepted approximation. As a result the congestion toll is given by:

$$r = \frac{c}{V(q)}\cdot\frac{\ln(V_f) - \ln(V(q))}{\ln(V(q)) - \ln(V_0)} \qquad (7)$$

Assuming that European Euro/km rates (Theaa 2008) are also valid for Italy, we can take cost values reported in Table 3 as input and then derive the value of *c=1,08 €/km*. Moreover, in order to resolve Eq. 7, values for V_0 and V_f are needed; while the value of speed at free flow can be considered as the one presented in Table 1 (V_f*=122,8 km/h*), the inference of a proper value for V_0 needs more attention. The evaluation of a realistic V_0 value is where our STRATUNA model can help and, in fact, leads to cost forecasting through speed forecasting.

Table 3: Total of Standing Charges and Running Costs, Assuming 15000 km per Year

	Euro/km	Euro/hour
Petrol	0,112844	0,31090068
Tyres	0,00806588	0,0222226
Service labour costs	0,02184835	0,06019521
Replacement parts	0,01472219	0,04056164
Parking and tolls	0,01409571	0,03883562
Standing charges	0,21981479	0,60561986
Total	**0,39139093**	**1,07833562**

In fact, our model has the expressivity needed for speed forecasting and has exhibited a predicting reliability for different flow volumes even in its partially implemented version (detailed in Section 2). Therefore, it can be used, together with the cost system object of this section, to foresee how different highway designs influence the speed at maximum capacity (V_0). This enables a straightforward calculation of the corresponding income for the highway owners and for the society. We now present the curves of *AC* and *MC*, as stated by Eqs. 3,4, with the aim of fixing the cost system.

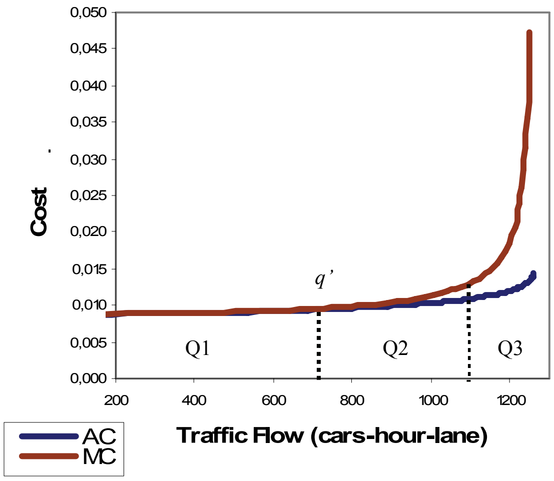


Figure 9: Cost of *AC* and *MC* in Relation to the Flow *q*

Up to a traffic volume of about 680 cars per hour per lane, the private cost of a motorist (*MC*) is, in fact, identical to the one that he imposes to others (*AC*). This, presented in Figure 9 as *Q1*, can be traced back to the free flow condition; same tracing is possible from *Q2* ad *Q3* (Cfr. Figure 9) to moderated flow and traffic jams, respectively. For quantity of cars $q > q'$ we have *AC* costs that increase more rapidly than *MC*: first linearly and then exponentially. This increasing cost, induced to others with heavier flow, can be represented by Figure 10: more cars means slower speed, that means more breaking/accelerating, low gears usage, higher petrol consumption and so on.

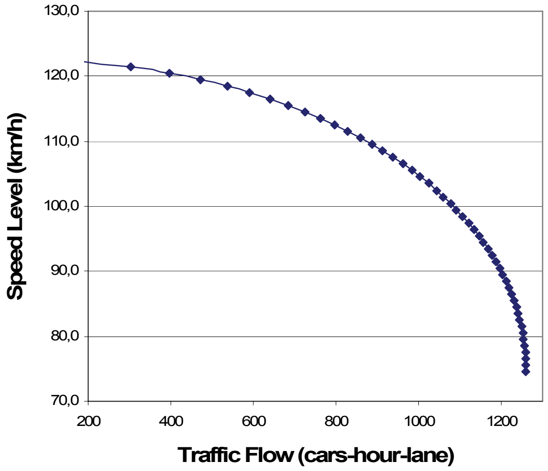


Figure 10: Speed-Flow Chart

Now that the cost system has been satisfactory detailed, we propose the congestion toll (€/km) evolution, in relation with the V_0 value deduced by our model when feed with scenarios detailed in section 2.

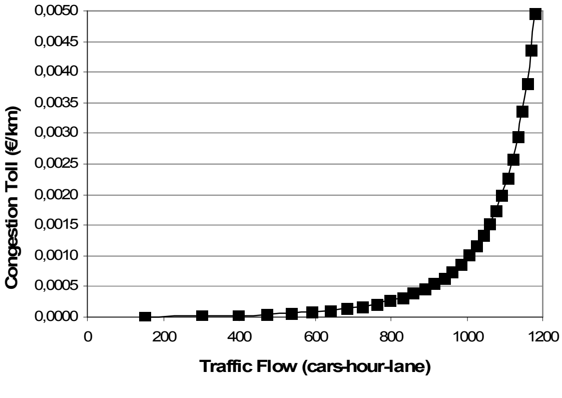


Figure 11: Congestion Toll with respect to Traffic Flow

Above results show clearly that, through a simulation model, the test of different highway designs is possible and then, to each design, is associable a simulated V_0 value, leading to the appropriate congestion toll. In other words, through the simulation of different highway design, differentiated V_0 values follow; then, the optimal congestion cost is derivable from it by means of the reported congestion toll system. As a result, we report in Table 4 the congestion toll that the price system of the simulated and analyzed highway could implement in relation to free flow, moderated flow and traffic jams.

Table 4: Congestion Toll and Different Traffic Flows

Flow type	Free flow	Moderated flow	Congestion
Corresponding minimum flow (cars per hour per lane)	0	681	1101
Corresponding maximum flow (cars per hour per lane)	680	1100	1260
Minimal congestion toll (€ per car per km)	+0,0	+0,0001	+0,0023
Maximal congestion toll (€ per car per km)	+0,0	+0,0023	+0,0339

5. CONCLUSIONS

A Cellular Automata approach for traffic flow modelling is proposed in this paper. The partial implementation $\beta 4$ presents interesting results in traffic forecasting task. Based on a validation on real data of the implemented model, the design has moved forward a more general CA simulator: the STRATUNA general model, created on the analysis of weak and strong points in implemented model along with the intent of an extended expressivity power. Moreover, when the general model implementation will be tackled, it will be possible to couple Genetic Algorithm to the simulator in order to fix information missing from the reality. As a result, the general model exhibits forecasting tools that, theoretically, outperform the mere speed prediction obtained with the preliminary implementation. The implemented model, used together with an established cost system, guides the interesting problem of the appraisal of the right price for a toll ticket. Indeed, the simulator shows the ability of associating to a simulated highway a value of average speed at maximum capacity. Thanks to this value, it is possible to establish a congestion toll mechanism. This mechanism, widely used worldwide, gives to motorists the perception of the costs they are imposing to others travelling and non-travelling people.

In conclusion, the CA approach demonstrates its validity and leads to interesting emerging phenomena, both from the traffic forecasting and from an economical point of view, where the model gives a feedback that straightforward links different highway designs to different congestion toll charges through an established cost system. In order to proceed to a further model improvement through validation, it is now important to get access to other types of data concerning highway traffic (i.e. single vehicle data), which authors are looking for at present.

REFERENCES

D'Ambrosio, D., Spataro, W., Iovine, G., 2006. Parallel genetic algorithms for optimising cellular automata models of natural complex phenomena: an application to debris-flows. *Computers and Geosciences*: 32. 861-875.

Di Gregorio, S., Festa, D.C., 1981. Cellular Automata for Freeway Traffic, *Proceedings of the First International Conference Applied Modelling and Simulation*, Vol. 5, pp. 133-136, September 7-11 1981, Lyon (France).

Di Gregorio, S., Serra, R.,1999. An empirical method for modelling and simulating some complex macroscopic phenomena by cellular automata. *Future Generation Computer Systems* 16: 259-271.

Di Gregorio, S., Festa, D.C., Rongo, R., Spataro, W., Spezzano, G., Talia, D., 1996. A microscopic freeway traffic simulator on a highly parallel system. In: D'Hollander E.H., G.R. Joubert, F.J. Peters, D. Trystam, ed.s *Parallel Computing: State-of-the-art and Perspectives*. Amsterdam: North Holland, 69-76.

Holland, J.H., 1975. *Adaptation in Natural and Artificial Systems*. Ann Arbor: University of Michigan Press.

Knospe, W., Santen, L., Schadschneider, A., Schreckenberg, M., 2000. Towards a realistic microscopic description of highway traffic. *Journal of Physics A-Mathematical and General*. 33: 477-485.

Làrraga, M.E., del Rìob, J.A., Alvarez-Icaza L., 2005. Cellular automata for one-lane traffic flow modeling, *Transportation Research Part C* 13: 63-74.

Li, M.Z.F., 2002. The role of speed–flow relationship in congestion pricing implementation with an application to Singapore. *Transportation Research Part B*: 36. 731-754.

Nagel, K., Schreckenberg, M., 1992. *Journal de Physique I* 2: 2221-2229.

Pigou, A.C., 1920. *The Economics of Welfare*. London: MacMillan.

Schadschneider, A., 2006. Cellular automata models of highway traffic. *Physica A* 372: 142-150.

Theaa, 2008. *Running Costs for Petrol Cars*. The Automobile Association Limited. Available from: http://www.theaa.com/allaboutcars/advice/advice_rcosts_petrol_table.jsp [Accessed 10 April 2008].

von Neumann, J., 1966. *Theory of Self Reproducing Automata*. Champaign: University of Illinois Press.

Wolf, D.E., 1999. Cellular automata for traffic simulation, *Physica A* 263: 438-451.

A FORMAL APPROACH FOR OPTIMIZED SYSTEM ENGINEERING

Yann Pollet[a], Olfa Chourabi[b]

[a] CNAM/CEDRIC, Paris, France.
[a]pollet@cnam.fr, [b] olfa.chourabi@voila.fr

ABSTRACT

System Engineering (SE) is a multidisciplinary activity in witch a system engineer team aims at building a global technical solution in response to a problem stated as a set of requirements. SE may be seen as process of optimizing some system quality factors under constraints of required functions, cost and delay. Complex products and systems, involve a large number of components which are arranged under constraints/relationships in engineering space. This may lead to uncontrolled and unpredictable backtracks in the engineering processes. Effectively managing such process is a major factor witch can increase product development capability and quality and reduce the development cycle time and cost. In this paper, we propose a formal approach for optimizing component allocation and parameter configuration in SE process. We introduce the concept of Alternative Quantity (AQ) to enable manipulation and reasoning on partial engineering choices. The main contribution is to offer efficient evaluation of global systems proprieties although particular technical choices have not been done yet.

Keywords: System Engineering Processes, Concurrent Engineering, Optimization, Product configuration

1. INTRODUCTION

System engineering (SE) is an interdisciplinary approach to enable the realization of successful systems. It is defined as an iterative problem solving process aiming at transforming user's requirements into a solution satisfying the constraints of: functionality, cost, time and quality [1]. This process is usually composed of the following seven tasks: State the problem, Investigate alternatives, Model the system, Integrate, Launch the system, Assess performance, and Re-evaluate. These functions can be summarized with the acronym SIMILAR: State, Investigate Model, Integrate, Launch, Assess and Re-evaluate. [2][3].

This process is shown in figure 1. It is important to note that the Systems Engineering Process is not sequential. The tasks are performed in a parallel and iterative manner. At each step a comprehensive set of possible engineering models arises witch are progressively combined and refined to define the target system.

Many engineering domains involve an intricate interplay of conceptual synthesis of alternative requirements and design configurations, preliminary impact analysis of these alternatives using complex simulations tools, and human decision-making. [4]

This may be seen as a process of optimising some system quality factor under constraints of required functionality, cost and delay. In current practices, SE activity is a decomposition of an initial problem into a hierarchy of sub problems, followed by design decisions taken for each elementary problem. A decision often consists in the choice of a given solution among n possible ones (e.g. the allocation of a required function to a given off-the-shelf component). As sub problems are not necessary independent, this leads in general to suboptimal solutions issued from putting together local decisions. In addition, multiple iterations may lead to uncontrolled and unpredictable backtracks in the engineering process. Concurrent Engineering shows situations in which all aspects of the system lifecycle are expected to be optimised together.

In this paper, we focus on decisions related to component allocation choices and parameter configuration, which may be formalised as optimisation problems under constraints. To fully evaluate the system and ensure its compliance to constraints, the system needs to be tested in a large number of crash scenarios. System performance is usually assessed using simulations. Therefore, computer modeling and simulation are used as one of the primary tools in the process. The number of potential combinations of alternative solutions is almost countless and the items to be considered are very closely related.

Effectively managing such process is a major factor witch can increase product development capability and quality and reduce the development cycle time and cost. We aim to present an original method to reduce the complexity of the system evaluation process. We propose an approach based on the well formalised concept of Alternative Quantity (AQ). An AQ offers a structure that enable the evaluation of global system properties while accurate design choices have not been done yet. The benefit is to reduce the number of simulation and to help to border the solution space.

Formally, an AQ represents a particular set of possible values from a D domain, e.g. a continuous domain such as the set of real numbers or a discrete domain such as a set of states of a sub system, each of one being associated to a specific hypothesis. AQ are distributions of a confidence measure over a D domain, where the values belong to a Boolean algebra of hypothesis, instead of being a real number between 0 and 1. In the case of the real number domain, it is possible to build algebra of AQ having calculation properties similar to those of classical real numbers. This leads to an easy transposing of classical calculus on AQ, enabling the evaluation of global systems properties although particular technical choices have not been done yet.

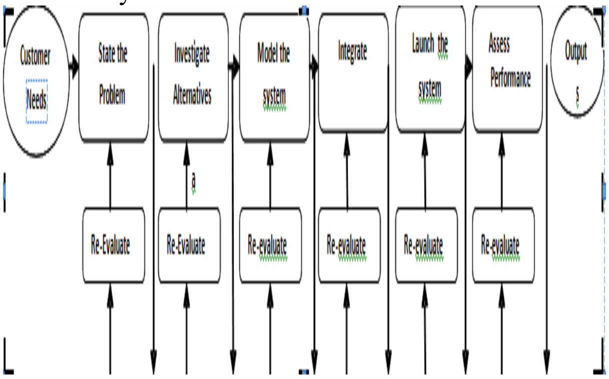


Figure 1. System engineering process

The paper is structured as follows. Section 2, discusses key background information about System Engineering processes and describes a motivating research example. Section 3, presents an overview of some related approaches for concurrent system engineering processes optimization. Section 4, defines Alternative Quantity (AQ) formalism with its underlying reasoning mechanism. In section 5, we evaluate our approach through an example of an automated transport sub system. The paper concludes with an overview of our contribution and future work directions

2. BACKGROUND AND MOTIVATIONS

In this section, we focus on decisions related to component allocation choices and parameter configuration phase in SE process. In this setting, the main objective is to find configurations of parts that implement a particular function. Practically, system engineer team must consider various constraints simultaneously, the constituent part of a system are subject to different restrictions, imposed by technical, performance, assembling and financial considerations, to name just a few. Combinations of those items are almost countless, and the items to be considered are very closely related. Figure 2, shows the intricate interplay of constraints in a system engineering process.

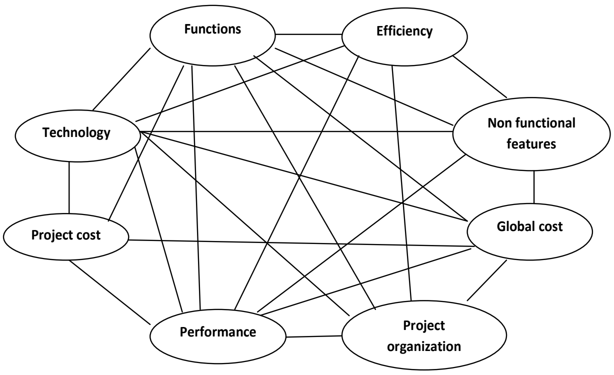


Figure 2: System Engineering process as a multi-objective optimization problems

As an example, we consider a typical component allocation process of a transportation sub system: an automated wagon.

We assume that the sub system's functional view is represented by the data flux diagram depicted in figure 3. The main functions considered are: F1: capture speed, F2: capture position, F3: control movement, F4 propel, F5: break, F6: contain travelers.

The system functions should be mapped to the physical components. Functions mapping to physical components can be one to one or many to one. This functional view was derived from the initial functional requirement: to transport travellers from one point to another. In addition Physical solution is constrained with non functional requirements (or soft goals) such as: system performance with attributes of travel duration, facility, acceleration limitation, comfort, reliability etc.

A possible physical solution for this system need to implement functions set, under constraints of non functional requirements and constraint of cost and leads time. Concurrently, models should be constructed and evaluated, simulation data should be derived, and prototypes should be built and measured.

In addition, it is important to emphasize that non functional requirements are reflected on allocated component in multiple ways. We could distinguish direct and indirect (or composite) requirement allocation. In our example, the attributes space and quality are directly assigned to the component passenger's cells, whereas, system weight attribute should be shared out among the system constituents. Others constraints impacting the global system and needs to be checked by simulation, such as efficiency of transport duration in different scenarios, system reliability etc.

Furthermore, each physical constituent choice raises a set of possible engineering alternatives. The global requirements are traded-off to find the preferred alternatives. An intricate interplay usually exists among alternatives. For example, the functions speed capture and position estimation choosing inertial station that delivers the speed as well as the position, for implementing the function speed capture would restrict the engineering choices to exclude specific transducers. Likewise, interplay may concern a common parameter. We could consider as a sample the dependencies between efficiency time parameter and the aggregate system weight parameter. We should note here that, in general, a subset of requirement is “rigid” i.e imposed and other is more flexible.

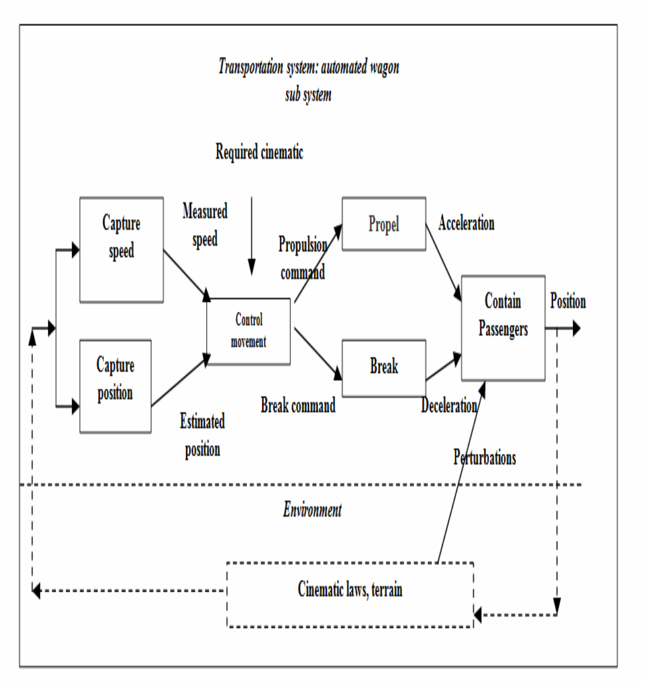


Figure3. Data flow diagram

The automated wagon example highlights multiple dependencies between technical solutions and mutual influences and trade off that arises in a quasi simplified use case. In system engineering practices, we face more complex interplays among local choices relating to sub system and global engineering decisions relating to the whole system.

Broadly speaking, the Re-evaluate loops depicted in figure.1 are arguably the most time consuming activities in System Engineering processes. Engineers have used feedback to help control systems and improve performance. It is one of the most fundamental engineering tools. Re-evaluation should be a continual process with many parallel loops. Re-evaluate means observing outputs and using this information to modify the system, the inputs, the product or the process.

It would clearly be useful to have some way of managing the multiple dependencies and exclusions between hypothetic alternative engineering choices, while saving traceability links and justifications.

3. RELATED WORK

In this section we examine some approaches to support decision making in system engineering processes. We discuss truth maintenance systems and constraint satisfaction problems approaches and we explain their inadequacy in our context. Then we explain our proposition of alternative quantities (AQ) informally, by the means of some examples. These examples are representative for the new structure we are putting forward, depicting different possible practical engineering use of AQ.

-Truth Maintenance Systems (TMS) / Assumption truth maintenance Systems (ATMS): Truth maintenance (also called belief revision or reason maintenance) is an area of AI concerned with revising sets of beliefs and maintaining the truth in the system when new information contradicts existing information. It is a subsystem that manages the use of assumptions in the reasoning process of a problem solver. Doyle's [8] original motivation for creating a truth maintenance system was to augment a reasoning system with a control strategy for activities concerning its non monotonic state of beliefs. These systems can be viewed as constraint propagation mechanisms. Given a disjunctive set of sets of premises and a set of (monotonic) deductive constraints, de Kleer's ATMS [9] [10][11] tells a client problem solving system what things it is currently obliged to believe, assuming one or another of the sets of premises. Doyle's and Goodwin's TMS's, on the other hand, tell the client problem solving system what things it is currently obliged to believe, given a single set of premises under deductive constraints, some of which may be non monotonic in nature.

Typically, complex engineering artifacts, involves successive choices and multiple iterations with backtracks in case of facing an unsatisfied constraint or requirement. But this process, of managing a complex network of assumptions is very time consuming and difficult to apply in practice. In addition, the demands on the development of systems are steadily increasing: shorter time to market, better quality, and better productivity are critical for the competitiveness of today’s SE organizations. For this reason, we argue that TMS/ATMS are not adequate in our context.

-Constraints satisfaction problems (CSP): A general introduction to constraints is found in [5]. Constraint-satisfaction problems can be stated as follows: We are given a set of variables, a finite and discrete domain for each variable, and a set of constraints. Each constraint is defined over some subset of the original set of variables and limits the combinations of values that the variables in this subset can take. The goal is to find one assignment to the variables such that the assignment satisfies all the constraints. In some problems, the goal is to find all such assignments. [6][7]

The major advantage of this paradigm is its high declarativity, domain independence and simplicity of use. Nevertheless, this formalism is not powerful enough to capture or to take advantage of essential aspects of system engineering practices, such as the unknown a priori number of constituent parts of a system, the unknown choice of a particular physical component and so on. All variables must be specified in advance, making it awkward to represent problems witch requires simulation results as inputs in order to keep or eliminate alternative solutions.

In such setting, we need to manipulate parameters that have not been totally defined. To address this modeling need, we introduce the concept of Alternative Quantity (AQ) [12]. AQ may be considered as structure encapsulating a set of possible values, enabling reasoning and calculation facilities with sound results, exactly as the case of well defined parameter.

Let's consider for example the case of the automated wagon. Let's assume that we haven't yet chosen the propel mode while we need the constituent weight in order to conduct simulations. Or, let's assume that we haven't yet fixed the transducer component while we need to have preliminary results to validate an engineering choice. In current practices, system engineers must simulate the global solution in order to assess system performance. As simulations require specific parameters this leads to multiple trial and error process before exhibiting an optimal solution.

With Alternative Quantities we are able to express that some variables can be assigned to a sub problem, rather than a simple value. In this way, components can be selected at a higher level, before being specified in terms of subassemblies. An AQ allows to encapsulate sets of possible values and to combine them soundly with the remaining system elements such as physical relations. Possible AQ values could be numeric or discrete, e.g. system state, as well

4. ALTERNATIVE QUANTITY: FORMAL DEFINITION

In order to express that a quantity is able to take multiple possible values in an engineering process, on develop an alike theory to a distribution of possibilities over a domain of values, called fuzzy quantity [13][14]. Possibility theory is a mathematical theory for dealing with certain types of uncertainty and is an alternative to probability theory. Zadeh [13] introduced possibility theory as an extension of his theory of fuzzy sets and fuzzy logic. D. Dubois and H. Prade [15] further contributed to its development.

4.1 Formal definitions

Distribution of possibilities (reminder)

Let Q be a domain of values discrete or continuous. A distribution of possibilities over Q is defined by a function μ mapping each element of Q into a closed unit interval [0, 1]. [15][16]

$\forall q \in Q: q \rightarrow \mu(q) \in [0,1]$

$\exists\, q0;\ \mu\,(q0) = 1$

A fuzzy quantity may be noted: $Q= \mu1/q1 + \mu2/q2 + \ldots + \mu n/qn$. Where qi denotes the possible values and μi denotes the values of corresponding possibilities.

For instance: the quantity Q= 1/1.2 + 0, 5/1.3 embody a real number having two possible values 1.2 and 1.3.

Certain operations of ordinary algebra, in particular addition (x+y), multiplication (xy), and construction of inverse (-x), could be extended and applied to fuzzy quantities. But we notice that basic proprieties of these operations are lost, e.g. associativity, inverse and distributivity.

For example, let q1, q2, q3 be fuzzy reals. We have: $(q1+q2)*q3 \neq q1*q3 + q2*q3$. The associatively propriety is not verified.

E.g. if q1, q2, q3 embody respectively two possibilities, then (q1+q2)*q3 is evaluated to 8 possibilities where as q1*q3 +q2*q3 is evaluated to 16 possibilities. In other term, the result depends on the operand order.

This example highlights a significant inadequacy to the applicability of this theory to our approach. A fuzzy quantity doesn't keep trace of the rationale underlying its calculation process. Thus, we can't have the same result as the sum of real number. This leads to limited applicability in a simulation context because we would like to be able to reason over possible values.

We introduce the concept of Alternative Quantity (AQ) as an extended fuzzy quantity that satisfies the following requirements:

It keeps traceability to its operand origin

Adapts possible values depending on engineering results.

Following, we present a formal definition to Alternative Quantities.

Definition 1: Alternative Quantity (AQ)

An alternative quantity with values belonging to a domain Q, is defined by a function f from Q to Ω, where Ω is a Boolean algebra such that:

The cardinal of Ω is infinite

There exists an infinite set of element ≠ {}, Xi (i=1, 2, …) such that :

If $i \neq j$

then $Xi.Xj = \{\}$, $Xi.\neg Xj = \{\}$, $\neg Xi.Xj = \{\}$, $\neg Xi.\neg Xj = \{\}$.

Xi (i=1, 2, …) are called principal elements or generators of Ω.

The elements of Ω expressed as the multiplication of principal element by its negative element are called irreducible.

Definition 3: Alternative Quantity Measure
An AQ measure is defined by a function mapping each pair (Xi,⌐Xi) to a pair of values belonging to [0,1].
(Xi,⌐Xi) → (μi, □i) ∈ [0,1]
where μi= 1 or □i=1.
μi is called measure of principal element Xi,
□i is called measure of negative element ⌐Xi.
We note |x| the measure of X.
With Alternative Quantity measure we are able to define a more expressive measure, which holds parameter values and their definition origins as well.

Definition 4: Alternative Quantity with finite number of values.
An AQ is said to be AQ with finite number of values if only a finite number of values from Q have an image different from the empty set. It could be expressed as:
X = X φ(1) /v1 + ….. X φ(n) /vn. Where
X φ(i) denotes the irreducible elements.
The support of X is ∪ X φ(i). If the support of X is Ω , then X is called always defined.
Example: alternative quantities with values in ℝ with finite number of values and always defined
X = X1/ 1 + ⌐X1/ 2. X is an alternative quantity evaluated to 1 or 2 and keeps the origin of its basic hypothesis X1 or ⌐X1 in simulation operations.
In addition, formal operations allow us, in practice, to manipulate alternatives quantities. For example, let Y= X2/ 2 + ⌐X2/5, be an AQ. We could calculate the sum of AQ (X+Y) by reducing to a common factor as following:
X+Y = X1.X2 / 1 + X1.⌐X2 / 5+ ⌐X1.X2 / 4 + ⌐X1.⌐X2 / 10
We could implement operations on QAR, by linking them to a mathematical library of alternative quantities. i.e. in object oriented programming, a QAR represents an instance of a class Alternative(float).

Interpretation: we propose to use AQ as a structure that embody assumptions hypothesis and choices at a step of an ongoing engineering process.

Principal elements Xi are used as symbolic elements to represent the set of all independent assumptions considered by system engineers until choosing the final engineering solution.
In an engineering situation Ω corresponds to an infinite set of assumptions. But in practical setting Ω is rather a finite set that represents engineer's assumptions underling different engineering choices.
Assumption measure |Xi| corresponds to the a priori acceptance of that assumption. This measure enables to assign preferences to alternatives solutions for an engineering problem.
The quantity |Xi| is used as a confidence measure for a particular belief.
Let's consider an engineering situation where two alternatives solutions are possible for a sub problem. We could put a superior confidence measure for the alternative that was used and verified in prior engineering situation. Else we could put 1 to mean that no preference is accorded.
In the example presented in section 2, the automated wagon consists of an inertial station and a passer cell. Each of these components can be represented by a variable having as domain a sub problem.
For instance the propel mode could have two values: petrol and electric power. We can assign a superior confidence measure to the solution electric power, if we have as soft goal environment conservation. This situation is formalized with an Alternative Quantity as following:

Propel mode = X1/petrol+ ⌐X1/ electric power with |X1|= 0,001 and |not X1|= 1

An alternative quantity is a parameter that enables to define a partially defined local solution. The major contribution is the facility of keeping different alternative element and reasoning on that structure as a well defined parameter i.e specific parameter value. An AQ could be all over defined if it s assigned to global engineering problem such as vehicle propel mode.
It could have limited scope if it is meaningful for a sub problem or dependent to specific choices. For example, lighting mode is meaningless if the propel mode is not petrol.

4.2 Real Alternative Quantity (QAR) algebra

This section describes an algebra for real alternative quantity i.e Q=ℝ. We argue that this algebra has calculation proprieties similar to those of classical real number. This leads to an easy transposing of classical calculus on AQ, enabling evaluation of global system proprieties although particular technical choices have not been done yet.

The binary operations: addition and multiplication are defined on QAR. Division is defined if the QAR has no possible null value.

QAR addition: commutativity, associativity holds for QAR. QAR addition has an identity element witch is zero.i.e QAR equal to zero for whatever hypothesis. Each QAR has an inverse element for addition.

QAR multiplication: commutativity, associativity and distributivity over addition. Each QAR has an inverse element for the product.

We present an example for QAR addition. This example highlights the transposing of classical calculus. In opposition to, fuzzy real where calculation is different from real calculation, QAR algebra gives similar results to those of classic real numbers. QAR is a structure encapsulating its possible values and the assumptions network on witch depend these values, as well.
For example,

Let's consider a fuzzy number having two possible values 1 and -1: x= 1/ (-1) + 1/ (+1)
Let's consider its inverse element for the addition: -x= 1/ (+1) + 1/ (-1),
The addition (x+ « -x ») gives three possibilities which are: -2, 0 and 2. This result contrasts with classical real calculus.
i.e: (x+ « -x ») must be evaluated to zero.
On the contrary, with QAR, we have the following results:
The same fuzzy number is expressed as:
x = X1/ (-1) + ⌐X1 / (+1) has as inverse element for the addition: x' = X1/(1) + ⌐X1 / (-1)
The result of (x + x') is evaluated to X1/0 + ⌐X1/0 = Ω /0 = 0, witch is conform to classical calculus of real.

4.3 Real Alternative Quantity (QAR) comparators

We could define a predicate θ on two QAR

θ:X, Y → {true, false}, where θ is a relation such as equality, inequality , superior to etc.

Let's present some examples for QAR comparison
Let x be a QAR, having value x= X1/0.8 + ⌐X1/1.1. (x < 1) is evaluated to X1/true + ⌐X1/false
Let x,y,z be QAR, having respective values :

x = X1/0.8 + ⌐X1/1.1
y = X1/0.9 + ⌐X1/1.0
z = X1/1.2 + ⌐X1/1.3

(x < y) is evaluated to X1/true + ⌐ X1/false

(x < z) is evaluated X1/true + ⌐X1/true = □/true = true

Let x, y be QAR having respective values

x = X1/0.8 + ⌐X1/1.1
y = X2/0.9 + ⌐X2/1.0

(x<y) is evaluated to X1.X2/true + X1.⌐X2/true + ⌐X1.X2/False + ⌐X1.⌐X2/False
= X1. (X2 + ⌐ X2)/true + ⌐X1. (X2+ ⌐X2)/False
= X1. Ω /true + ⌐X1. Ω /False
= X1/true+ ⌐X1/False

This result is intuitive. In fact, (x<y) is dependant of the hypothesis X1. The comparison result sets that if X1 is true then (x<y) else the result is evaluated to false.
We could define an algorithm to automate the testing process of QAR. This would enable to border hypothetical solution space providing a set of constraints. Following a sample illustrating this principle:
x = X1/0.8 + ⌐X1/1.1
y = X2/0.9 + ⌐X2/1.0
if (x < y) then x = x + 1
(x < y) is evaluated to X1/true + ⌐X1/False
(x < y) is supported by X1
x is evaluated to x+1 = x + Ω /1 if X1is true and remains if ⌐X1is true
x = X1.(X1/1.8 + ⌐X1/2.1) + ⌐X1.(X1/0.8 + ⌐X1/1.1) = X1/1.8 + ⌐X1/1.1

5. APPLICATION: MODELING A TRANSPORTATION SUB SYSTEM WITH ALTERNATIVE QUANTITY

As mentioned in the previous section, the AQ represents particular sets of possible system proprieties values associated with underlining hypothesis at a given step of an engineering process. This section details shows how AQ can be effectively used to support system evaluation and verification process.
We use the case study presented in the motivation section to explore the advantages that can be derived from the use of AQ. The study shows how, by employing such a modeling structure, it is possible to reduce the complexity of the simulation process.
The transportation sub system is a self piloted wagon. We focus on the synthesis phase and we discuss decisions related to component allocation choices and parameter configuration. Figure 4 presents an allocation sample for the automated transport system. The set of parameter for the physical solutions are presented.

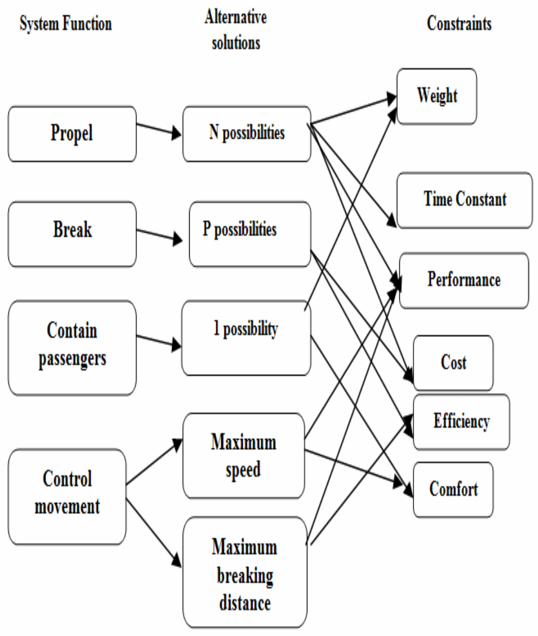


Figure. 4 allocation process sample for an automated transport system

Alternative designs are created and are evaluated based on performance, schedule, cost and risk. Then, concurrently, models should be constructed and evaluated, simulation data should be derived and prototypes should be built and measured.

Let's consider for example the following simulations scenarios:

-Control component (a software in an embedded calculating): having as
Inputs: measured speed and measured position.
Outputs: acceleration control and breaking control
Parameters: maximum speed, maximum acceleration and maximum breaking distance (with values within a possible interval)
-Engine component: having as
Inputs: acceleration control
Outputs: acceleration
Parameters: engine type
-Braking component: having as
Inputs: deceleration control
Outputs: deceleration
Parameters: breaking component type
-Vehicle Kinematic: without parameters as the vehicle is already chosen at this stage.
-Transducers: that are assumed to be perfect, characterized by: measured speed = real speed and measured position = real position.

Concerning the control, we have a succession of four phases corresponding to different system states. In each phase, we consider a defined control law, witch controls the acceleration in the phase acceleration, controls the acceleration and the braking in the phase constant speed and controls the braking in the phase deceleration. Law successions could be modeled as a state machine diagram, where the transitions could be either events e.g. departure or predicates on reached speed, reached position etc. e.g.(acceleration phase transition-deceleration phase) or (constant speed transition-deceleration phase). A new system state is a function of current state and input. i.e. state n+1= F (state n, input).

For the vehicle we use the kinematic equations e.g.
$S(n+1).dt = S\ n.dt + \gamma.dt$
$P(n+1).dt = P\ n.dt + S\ n.dt$

Where dt denotes the time pace of the simulation and γ denotes the acceleration or the deceleration.

For the engine, we use an approximation By a first order transfer function with a delay T0Engine. This function gives the following recurrent formula: $\gamma(n+1).dt = A.\gamma\ n.dt + B.Un.dt$.

Where u n.dt is the acceleration control at the instant n.dt

A is fixed by the formula: $e^{-T0.dt}$ and $B = 1 - A$.

In our approach, the classical formula, detailed above, we substitute the classical reals values or the system state with real alternative quantities or by alternative states quantities.

We have implemented this approach, by developing a mathematical library with basic alternative quantities classes. Each alternative object is an AQ class instance representing a set of alternative values. The operators, detailed in section 6, e.g. addition, multiplication, inverse, and QAR comparators are used to evaluate alternative results. In this way, alternative quantities allow to considerably optimize and reduce the complexity of the simulation process.

CONCLUSION

During component allocation process, system engineers typically have to find configurations of parts that implement a particular function. This process is constraint orientated, and requires the recognition, formulation and satisfaction of constraints.

In this paper we have introduced a novel formal approach to support efficient product configurations. We have discussed the contribution of alternative quantities i.e. to offer sound reasoning mechanism on product parameter although particular technical choices have not been done yet. We are currently investigating application of the proposed formalism in real engineering projects.

REFERENCES

J.Meinadier. Le métier d'intégration de systèmes. Hermes Science Publications, décembre, 2002.

A. T. Bahill and B. Gissing, Re-evaluating systems engineering concepts using systems thinking, IEEE Transaction on Systems, Man and Cybernetics, Part C: Applications and Reviews, 28 (4), 516-527, 1998

International Council on Systems Engineering (INCOSE). [On line] www.incose.org

Shinya Tarumi et al. Development of a Design Supporting System for Nano-Materials based on a Framework for Integrated Knowledge of Functioning-Manufacturing Process. In Proc. of the 10th IASTED International Conference Intelligent Systems and Control(ISC2007),Cambridge, Massachusetts, USA, Novemb 19-21,pp.446-454, 2007

V. Kumar. Algorithms for constraint-satisfaction problems: A survey. AI Magazine, pages 32 – 44, Spring 1992

Freuder, E. 1989. Partial Constraint Satisfaction. In Proceedings of the Eleventh International Joint Conference on Artificial Intelligence, 278–283.

Gu, J. 1989. Parallel Algorithms and Architectures for Very Fast AI Search. Ph.D. diss., Computer Science Dept., Univ. of Utah.

Jon Doyle. Truth Maintenance Systems for Problem Solving. Technical Report AI-TR-419, Massachusetts Institute of Technology, Artificial Intelligence Laboratory, Cambridge, January 1978

de Kleer, J. 1986a. An Assumption-Based TMS. Artificial Intelligence 28:127–162

de Kleer, J. 1986b. Problems with ATMS. Artificial Intelligence 28:197–224

de Kleer, J. 1989. A Comparison of ATMS and CSP Techniques. In Proceedings of the Eleventh International
Joint Conference on Artificial Intelligence, 290-296. Menlo Park, Calif.: International Joint Conferences on Artificial Intelligence

Sébastien ROBIDOU, Yann POLLET. Imprecision and Uncertainty in Object Oriented Databases. 5th European Congress on Intelligent Techniques & Soft Computing, EUFIT'97. Aachen, Germany, September 1997.

L.A. Zadeh (1965) Fuzzy sets. Information and Control 8 (3) 338-353

Zadeh, L., Probability measures of fuzzy events, Jour. Math. Analysis and Appl. 23, 421-427, 1968.

Dubois and Prade, Possibility theory and data fusion in poorly informed environments. Control Engineering Practice, 2(5):811-823.

Duboi et al. Readings in Fuzzy Sets for Intelligent Systems. Morgan Kaufmann, 2929 Campus Drive, Suite 260, San Mateo, Californie 94403, USA.

Yann POLLET, Eric RICARD et Sébastien ROBIDOU. A fuzzy Spatio Temporal Data Model for CIS. In "Fuzzy Logic and Soft Computing", edited by B. Bouchon-Meunier, R. R. Yager and L. A. Zadeh. World Scientific, 1995.

Yann POLLET and Sébastien ROBIDOU. An approach for the representation of multi-valued attributes in Fuzzy Databases. FUZZ-IEEE/IFES'95 Workshop on Fuzzy Databases and Information Retrieval. Yokohama, Japon, mars 1995.

Yann POLLET et Sébastien ROBIDOU. STORM : une approche pour la prise en compte de l'imprécision et de l'incertitude dans la gestion des grands projets". 11ème Colloque National de Fiabilité et Maintenabilité (LambdaMu11), session "Mathématiques appliquées à la SdF". Arcachon, septembre 98.

HYDROLOGICAL RISK ANALYSIS OF SOLIMÕES RIVER USING EXTREME VALUE THEORY

Alexandra R. Mendes de Almeida[a], Beatriz V. de Melo Mendes[b], Maria Célia S. Lopes[c], Gerson Gomes Cunha[d]

[a], [c], [d] LAMCE/COPPE-UFRJ
[b] IM/COPPEAD-UFRJ

[a] alexrma@lamce.coppe.ufrj.br, [b] bmendes@visualnet.com.br, [c] celia@lamce.coppe.ufrj.br,
[d] gerson@lamce.coppe.ufrj.br

ABSTRACT

The Brazilian hydrography is made by naturally navigable rivers that, in most cases, are the only transport means between many villages located by their margins. It occurs in particular in Amazonas (Brazil), where large distances and adverse natural conditions, make difficult the use of other transportation types, whose implantation and use are too onerous.

Rivers are crucial not only for transporting large cargoes through large distances, they are also important to the local and international commerce, making viable the offer of products with competitive prices.

This work presents the study with extreme low levels of rivers at Amazon region. In this study, we consider specially Solimões River levels, because Solimões is very important in communicating regular lines of cargo and people fluvial transport between local villages.

The use of the Extreme Value Theory to model these extreme low levels of the Solimões river is due to its adequacy and accuracy in estimating probabilities of occurring rare events.

Keywords: fluvial transportation, extreme value theory, hydrology.

1. INTRODUCTION

The importance of Solimões river in Amazonas is because of its strategic geographic position, that makes it an attractive option to the products leakage, besides constitute a principal transportation way to the local people who lives at the margin of supplementary rivers.

The objective of this paper is to specify the occurrence of rare events that, in this context, are the low levels of the Solimões river.

In EVT we study the behavior of rare events, the less frequent events that in major cases have an unarguable impact, and on emphasizing events of less probability of occurrence we achieve a better accuracy in modelling the tail distribution of river levels.

We use Extreme Value Theory in modeling low levels of Solimões River, analyzing particularly six monitoring stations that are placed since the border of Brazilian territory until near its union with Negro River. They are: Tabatinga, São Paulo e Olivença, Santo Antônio do Içá, Fonte Boa, Itapeuá e Manacapuru.

First applications using EVT were in meteorological phenomena modeling area and involved maximum precipitation and annual level of inundation in United States. However, EVT approach is quite inclusive and can be applied in a variety of natural phenomena like: inundations, atmospherical pollution, engineering, actuary and financials.

2. EXTREME VALUE THEORY

Extreme Value Theory has emerged as one of the most important statistical disciplines for the applied sciences over the last 50 years, also becoming widely used in natural phenomenon like inundation, atmospheric pollution, engineering, level of hydroelectric pond (Almeida and Mendes 2005), actuary and finance.

The most important results was obtained in Fisher and Tippett (1928), Gnedenko (1943) and Galambos *et al* (1994).

The distinguishing feature of an extreme value analysis is the objective to quantify the stochastic behaviour of a process at unusually small levels. In particular, extreme value analyses usually require estimation of the probability of events that are more extreme than any that have already been observed.

The discussion about the better method in EVT to model the extremes comes from the quantity of data, considering that by definition, extreme observations are less frequency.

The EVT model that treat events that don't exceed a small threshold u is denoted by Generalized Pareto Distribution.

2.1. The Generalized Pareto Distribution

The Generalized Pareto Distribution (GPD) is an approximation to the limit distribution of excess smaller than a small threshold. We call *excesses* the values smaller than the threshold, and *excess* the difference between the excesses and the threshold.

Consider $X_1, \ldots, X_n$ a sequence of independent and identically distributed random variables, having marginal distribution function F. The extremes events are the X_i that exceed some high threshold u.

Denoting an arbitrary term in the X_i by X, the behavior of extremes events is given by the conditional probability:

(1)

If the parent distribution F is known, the distribution of threshold exceedances given above would also be known. Since, in practical applications, this is not the case, approximations that are broadly applicable for high values of the threshold are sought, Coles (2004).

The distribution function of the excess y is given by the GPD:

(2)

Where $\sigma>0$, $y\geq0$ when $\xi\geq0$, and $0\leq y\leq -\sigma/\xi$, when $\xi<0$.

The parameters ξ and σ is related to the shape and scale of the model.

This is a generalized distribution, including other distributions above the same parametric function. If $\xi>0$ then H(y) is a re-parametrized version of the Standard Pareto Distribution, widely used in actuarial modelling of big losses. If $\xi=0$ we have the exponential distribution and, finally if $\xi<0$ we have the Type II Pareto distribution.

3. THE CASE STUDY

3.1. The Data

The data analyzed was made available by the Agência Nacional de Águas (ANA-Brazil), and consists in daily level (in meters) of six monitor stations distributed across the Solimões river, and is considered by ANA as of most strategic interest: Tabatinga, São Paulo de Olivença, Santo Antônio do Içá, Fonte Boa, Itapeuá and Manacapuru.

These stations were chosen in order to cover all the extension of Solimões River since the Brazilian boundary (Tabatinga e São Paulo de Olivença) passing through points at medium riverbed (Santo Antônio do Içá, Fonte Boa e Itapeuá), until the proximities of its union with Negro River. Figure 1 shows all the extension of Solimões River as well as the location of all stations used.

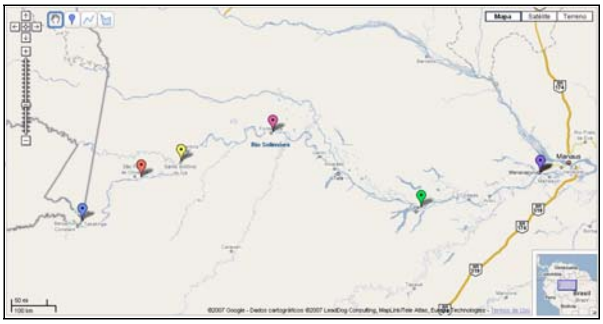


Figure 1: Monitor station's geographic localization in Solimões river.

The levels data series periods vary for each station. Tabatinga data series starts at july of 1982 and ends at November 2007, São Paulo de Olivença and Santo Antônio do Içá collected data from july 1973 until june 2007, Fonte Boa start at november of 1977 and ends at november of 2007, has data from april of 1971 until july of 2007, finally Manacapuru starts at june of 1972 and ends at January of 2007.

Environmental data are typically high correlated, and the daily levels presents short and long memory. The Ljung-Box test for serial correlation (Box and Pierce, 1970) and the R/S for long memory (Lo, 1991), reject their null hypothesis at 1% significance level. Figure 2 shows how strong is the daily levels autocorrelation function of Fonte Boa and Itapeuá monitor stations.

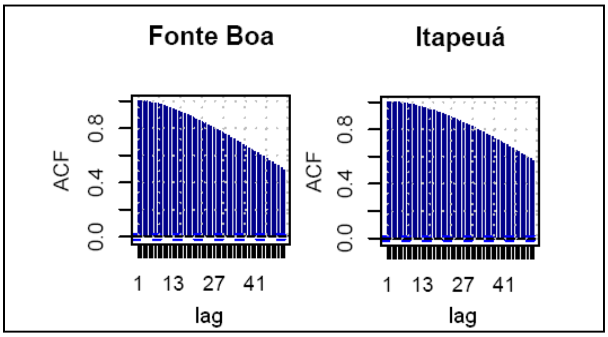


Figure 2: Autocorrelation function plot of Fonte Boa and Itapeuá monitor stations.

3.2. Monthly Minima

To remove the temporal dependency of the time series, we selected the monthly minimum of daily levels.

Another concern is about the trend and seasonality of the data. According to Figueroa and Nobre (1990) the spatial-temporal distribution of the precipitation, an important factor to determine the river level, is characteristic of each location. The Figure 3 shows the spatial variation of Amazonian precipitation:

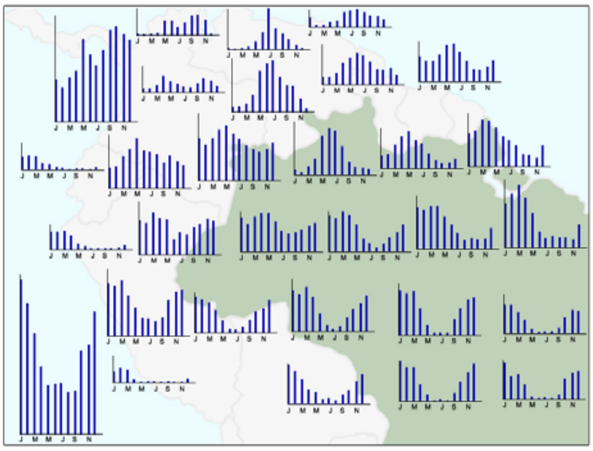


Figure 3: Amazonas spatial-temporal precipitation distribution.

We analyzed the levels profile of each monitoring station and considered spatial characteristics of Amazon region as shown in Figure 3. We verified a delay in water dam and decided to segment the data series in three distinct profiles: periods of dry, high and transition, enclosing different months for each station, once the Amazon region presents distinct precipitation indexes for locations not so far from each other.

Figure 4 shows box plot of levels separated by months of Tabatinga station, where blue color indicates the period considered as high, green color indicates months classified as transition period and red color indicates months of river dry.

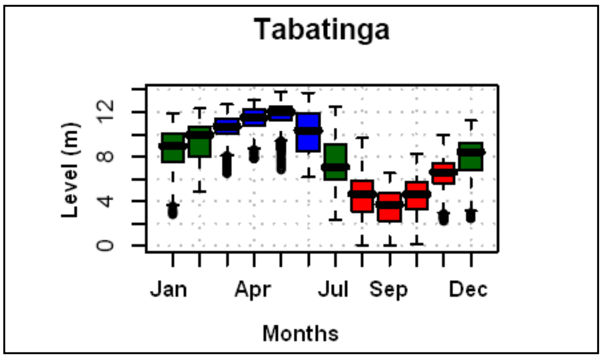


Figure 4: Box plot of monthly minima levels.

Although the monthly minima still have a weak dependency in short lags, like shown in figure 5, that can be explained by the kind of data; but we do not see long memory evidence anymore.

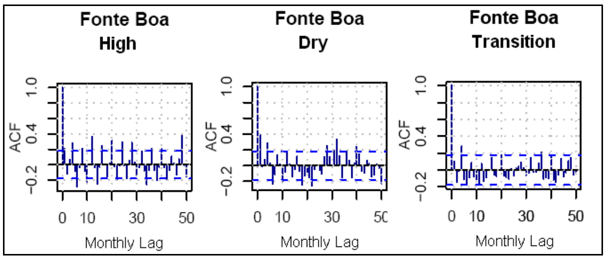


Figure 5: ACF for monthly minima levels by periods.

We conclude the data study verifying the series stationarity. The use the KPSS test, which the null hypothesis is stationariry (level and in tendency), and verify that for all monitor station and season, we could not reject the null hypothesis at in average, 5% significance level.

3.3. Modeling and Inferences

The choice of the threshold *u* is a critical step of the modelling. A too high threshold is likely to violate the asymptotic basis of the model, leading to bias; and a too low threshold will generate few excesses with which the model can be estimated, leading to high variance (Coles 2004).

We estimated the threshold *u* by a sensibility analysis, based on the fitting of models across a range of different threshold, resulting on the choice of the 21.5% quantile of each series/period.

The parameters of the GPD was estimated by a numerical method called *Probability Weighted Moments (Hosking 1985).*

Table 1 shows the threshold (u), the number of excesses N_u, and the estimated shape and scale, by period and monitor station.

Table 1: Estimates by period and monitor station.

Monitor Station	Period	u (cm)	N_u	ξ estimated	σ estimated
Tabatinga	High	841,84	22	-1,083	2,486
	Dry	216,01	23	-0,644	1,655
	Transition	492,00	21	-0,013	0,984
São Paulo de Olivença	High	1033,98	30	-0,374	1,849
	Dry	387,07	30	-0,119	0,944
	Transition	681,48	29	-0,040	0,993
Santo Antônio do Içá	High	1014,17	30	-0,041	1,520
	Dry	326,02	30	-0,226	1,294
	Transition	620,34	29	-0,026	1,052
Fonte Boa	High	1851,58	26	0,150	1,154
	Dry	1142,92	26	-0,197	0,946
	Transition	1462,58	26	-0,087	1,150
Itapeuá	High	1338,07	32	-0,079	1,199
	Dry	626,49	31	-0,278	1,856
	Transition	866,74	31	-0,491	3,179
Manacapuru	High	1594,92	30	-0,243	1,538
	Dry	833,96	30	-0,230	1,773
	Transition	1174,45	30	-0,320	2,524

The shape (ξ) parameter estimated of almost all station/period was negative, indicating the limitation of the series, as expected, and the scale (σ) parameter estimated was similar for the same periods of the stations.

To verify the goodness-of-fit of each model we used the Kolmogorov-Smirnov test, that basically compares the empirical distribution function with the cumulative distribution function specified by the null hypothesis. The results can be seen in table 2.

Table 2: Estimates by period and monitor station.

Monitor Station	Period	N_u	D_n	P-value
Tabatinga	High	22	0,13	87,38%
	Dry	23	0,11	92,18%
	Transition	21	0,12	86,17%
São Paulo de Olivença	High	30	0,07	99,69%
	Dry	30	0,13	65,77%
	Transition	29	0,12	77,53%
Santo Antônio do Içá	High	30	0,10	92,08%
	Dry	30	0,09	96,86%
	Transition	29	0,13	74,85%
Fonte Boa	High	26	0,14	71,61%
	Dry	26	0,09	98,11%
	Transition	26	0,11	93,54%
Itapeuá	High	32	0,07	99,55%
	Dry	31	0,09	96,04%
	Transition	31	0,15	44,70%
Manacapuru	High	30	0,08	99,38%
	Dry	30	0,16	40,12%
	Transition	30	0,11	85,31%

At the significance level of 5% we accepted all null hypothesis of goodness-of-fit.

4. RESULTS

Once the estimation are obtained and model adequacy is verified, we could use the model to obtain results that reflects estimation impact, which characterize the behavior of the model in a future scenery. An interesting analysis in this context of minimal levels of monitoring solutions of Solimões River consists in obtaining the events in adverse sceneries.

Frequently it is necessary to construct adverse scenarios to warn us of some extreme event. The best way to take care of this is to assign to this event an average time to occur.

In general, we impose a large period to study the value of the probability of its excess.

The return level of *t*-periods, as in Mendes (2004), is a concept that can be summarized basically by the association of a quantile of the excesses distribution and a expected period *t* between their occurrence.

The equation of return level of t-periods is given by:

(3)

Where x_t is the return level of *t*-periods, u is the threshold, ξ and σ are the shape and scale parameters, t is the number of periods, and ζ_u is the $P\{X<u\}$.

Table 3 shows the estimates of *t*-period return, for t=5, 10, 15 e 20.

Table 3: T-period return estimates.

Monitor Station	Period	*t*=5	*t*=10	*t*=15	*t*=20
Tabatinga	High	8,245	7,124	6,768	6,596
	Dry	2,043	1,159	0,798	0,594
	Transition	4,848	4,170	3,776	3,498
São Paulo de Olivença	High	10,207	9,108	9,585	8,260
	Dry	3,802	3,179	2,838	2,605
	Transition	6,743	6,066	5,678	5,407
Santo Antônio do Içá	High	10,031	8,995	8,403	7,988
	Dry	3,167	2,350	1,898	1,651
	Transition	6,127	5,405	4,989	4,697
Fonte Boa	High	18,431	17,579	17,038	16,634
	Dry	11,361	10,756	10,439	10,229
	Transition	14,542	13,773	13,344	13,049
Itapeuá	High	13,294	12,490	12,039	11,729
	Dry	6,131	4,984	4,409	4,039
	Transition	8,441	6,639	5,836	5,357
Manacapuru	High	15,838	14,874	14,381	14,060
	Dry	8,212	7,082	6,491	6,101
	Transition	11,564	10,030	9,279	8,802

The estimated return level of *t*-periods of each station shows that for 10 years period the high and transitions periods has a low estimates, enough to makes hydrological risk managers includes this statistics in their future decisions.

5. CONCLUSIONS

The Extreme Value Theory is a powerful methodology to model all nature of risks. In this paper we use this approach to estimate extreme low levels of the Solimões (AM-Brazil) river.

The calculus of more accurate estimates for the low levels of the river is important to choose the type (and size) of the ship for use in cargo transportation.

In fact the data segmentation in periods of high, dry and transition water results in robust statistics of return levels, that encourage us to study the relationship between the seasons of each monitor station, and mainly the lag of each extreme low level.

REFERENCES

Almeida, A. R. M. de, Mendes B.V.M., 2005. Modelagem de Eventos Extremos no Setor Hidrelétrico. *Iniciação Científica (UFRJ - CNPq).*

Coles, S. G., 2004. An Introduction to Statistical Modeling of Extreme Value. *Springer-Verlag.*

Embrechts, P., Klüppelberg, C. and Mikosch, T., 1997. Modelling Extremal Events. *Springer-Verlag.*

Figueroa, S. N., Nobre, C. A., 1990. Precipitation Distribution over Central and Western Tropical South America. *Climanálise, V. 5, N. 6.*

Fisher, R. A., Tippett, L. H. C., 1928. Limiting Forms of the Frequency Distribution of the Largest or Smallest Member of a Sample. *Proceedings of the Cambridge Philosophical Society, V. 24, p. 180-190.*

Galambos, J., Leigh , S. , & Simiu, E., 1994. Extreme Value Theory and Applications. Kluwer, Amsterdam.

Gnedenko, B. V., 1943. Sur la Distribution Limite du Terme Maximum d´une Série Aléatoire. Annales des Mathématiques, v. 44, p. 423-453.

Hosking, J. R. M., Wallis, J. R. e Wood, E. F., 1985. Estimation of the Generalized Extreme Value Distribution by the Method of Probability-Weighted Moments. *Technometrics, 27, 251-161.*

Mendes, B.V.M., 2004. Introdução à Análise de Eventos Extremos. *Editora E-Papers.*

EVENT SCHEDULING MADE EASY: BASIC SIMULATION FACILITY REVISITED

Luís S Dias [(a)], **Guilherme A B Pereira** [(b)], **José A Oliveira** [(c)]

[(a)(b)(c)] Departamento de Produção e Sistemas
Universidade do Minho
4710-057 Braga – Portugal

[(a)] lsd@dps.uminho.pt, [(b)] gui@dps.uminho.pt, [(c)] zan@dps.uminho.pt

ABSTRACT

In this paper, the use of Event Scheduling concepts for modeler-client communication, acting as an automatic generator of simulation programs under event scheduling paradigm - Basic Simulation Facility, thus eliminating any programming effort and expertise, is studied.

The main idea behind this work is to enhance the utilization of flowcharts in modeling, making it a great contribution to automatic generation of simulation programs, keeping it simple and portable.

The software tool developed is supposed to perform a sequence of what-if questions with a simple interaction with the user. These questions are then capable of automatically generating a program structure based on Event Scheduling simulation approach, using the Basic Simulation Facility routines.

This software tool would then contribute to the generalization and better understanding of the use of simulation as it only requires expertise in a basic simulation approach – Event Scheduling, thus incorporating simple flowcharts defining the system and its functioning rules.

Keywords: Event Scheduling, Simulation, Basic Simulation Facility, Automatic Generation of Simulation Programs

1. INTRODUCTION

Simulation is simply the use of a computer model to “mimic” the behavior of a complicated system and thereby gain insight into the performance of that system under a variety of circumstances (Thesen and Travis 1990).

In this paper, we keep the suggestion of using a (simple) graphical support as a representation of how the system really behaves, but these diagrams will also act as the source to the automatic generation of simulation programs.

In particular, “…Discrete event simulation models are run by tracing events over time, particularly those events that change the state of the system. Since we do not normally think of systems in terms of events and state changes, we usually use a simulation language or software package that allows us to represent the model more naturally. The computer then translates this to an event oriented approach (i.e., events and state changes) to actually run the model…” (Thesen and Travis 1991). The traditional approach for discrete event simulation modeling (Dias and Rodrigues 2002), includes visual support diagrams for modeler-client communication purposes (model interpretation and validation) and also to act as the basis for simulation language program construction.

Michael Pidd (1992), and Tocher (1963), even support that when generic programming languages were replaced by specific purpose simulation languages the use of paper diagrams remained as a previous step to programming.

The use of visual support diagrams to help the programming step of a simulation project is very common – this paper emphasizes this step, by proposing a way of automatically translating it into a simulation program. In fact, the abstraction of these diagrams serve as a support to the communication between the simulation client and the modeler (simulation expert), but also help the construction of the corresponding computational programs (Clementson 1982).

Dias and Rodrigues (2002), support that the new powerful graphical interfaces available with modern simulation languages (Dias, Pereira and Rodrigues 2007) are clearly programmer oriented, raising the difficulty in communicating with the client and still requiring enormous simulation expertise to use them.

The use of Event Scheduling concepts for modeler-client communication, acting as an automatic generator of simulation programs under event scheduling paradigm - Basic Simulation Facility (Thesen 1978), thus eliminating any programming effort and expertise, is studied.

The main idea of this work is to enhance the utilization of flowcharts in modeling, making it a great contribution to automatic generation of simulation programs, keeping it simple and portable.

Flowcharts are probably the former and most widely used graphical syntax in behavior specification (Gilbreth and Gilbreth 1921). The first known mathematical formalization was made by Nassi and Shneiderman (1973). It can be accepted as a universal visual language, and it is easy to assume that every professional, in some technical work, has already used it.

2. SOFTWARE TOOL DEVELOPED

The work presented in this paper could constitute a major step towards the generalization of the use of simulation. In fact, we suggest the use of a simple interface (Event Scheduling Flowcharts) to model a real situation. Then we present a tool capable of generating a simulation program. Based on event scheduling simulation modeling philosophy, our tool automatically generates a program to use Basic Simulation Facility routines. Furthermore the mentioned automatic generation of simulation programs does not require great expertise in simulation.

The Event Scheduling simulation philosophy is based upon the identification of events. An event corresponds to a point in time where there is potentially a modification on the state of the system under analysis. The identification of each event is complemented with the definition of the tasks to be performed each time an event occurs. These tasks would include:

1. Managing queues (either removing or inserting entities in queues)
2. Managing resources utilization (either seizing or releasing resources)
3. Recording statistics (for future evaluation of performance indexes, i.e., average waiting time in queue, average queue length, average resource utilization, etc.)
4. Generating random variables
5. Managing future events schedule

The definition of events above is very broad, promoting the construction of general models, hence promoting the automatic generation of simulation programs.

In fact, this broad explanation of an event and the above definition of tasks involved in each event made it possible to develop a set of general routines that apply to a large set of real problems, hence making adequate the use of this facility as far as the Event Scheduling approach is used. One of these facilities is the Basic Simulation Facility – BSF (Thesen 1978).

BSF includes four routines:

1. INIT – essentially dedicated to the design and initialization of the data structure that implements the simulation
2. INSERT – basically dedicated to the insertion of a record into a file (could be the implementation of an arrival to a queue, or seizing a resource or even the planning of a future event)
3. REMOVE – basically dedicated to the removal of a record from a file (e.g. the removal of an entity from a queue, or releasing of a resource or even getting information of a future event)
4. REPORT – essentially dedicated to the computation performance measures

These routines and the philosophy associated could be found (implemented) in various programming languages. Nevertheless, it is essential to develop a computer program, specifically dedicated to the system under analysis, that would invoke these routines, thus creating a mimic of the system.

This step is therefore a step that implies expertise in Event Scheduling simulation philosophy. This paper is then dedicated to present a tool that, based on the Flowcharts constructed for each of the events identified and based on a simple questionnaire related to some particularities of the Flowcharts, would automatically generate a program specifically developed to respond to a real system.

The software tool developed is supposed to perform a sequence of what-if questions with a simple interaction with the user. These questions are then capable of automatically generating a program structure based on Event Scheduling simulation approach, using the Basic Simulation Facility routines mentioned above.

The dialog with the user is based on four possible options:

1. Add Event
2. Remove Event
3. List Events
4. Compile

"Add Event" is the option responsible for adding an event to the model. The corresponding dialog incorporates knowledge on the type of event (whether corresponding to an arrival event or an event related to the use of a resource); on the random variable associated; on the number of resources available, if applicable; on the following resource used, if any, and also on the conditions that would permit that use.

"Add Event" is really the most important option. Here, we demonstrated what exact questions would simultaneously completely define each event, i.e., which questions would be sufficiently self-explanatory to fully define the important information related to each event and also integrate (coordinate) the follow-up, if any, to the use of the next resource.

"Remove Event" and "List Events" just intend to implement the basic options of respectively removing a previous defined event or listing events entered. The "Compile" option simply generates the program according to the answers to the what-if questions that really represent the system rules that will cause the mimic of the system and then compiles the program, enabling its execution. This phase would also create a text file as a result of the mentioned dialog with the user. This particularity of the software tool also permits that a user, well knowing the features of this file, could then simply carry out the previous definition of that file – this would enable to automatically generate the Basic Simulation Facility program without the necessity of executing the mentioned dialog. This represents an operation that requires a certain expertise in Event Scheduling philosophy but, nevertheless, the dialog is always a way to achieve the same result.

3. APPLICATION EXAMPLE

For a brief explanation of the tool developed, we present a simple system which could be represented by a simple simulation model following Event Scheduling approach.

The system incorporates the arrival of entities, according to time between arrivals described by an exponential type of distribution with mean 3 minutes.

Once in the system, the entities would require the use of a first resource (Resource_1). The time for Resource_1 utilization would follow a Random Distribution between 5 and 9 minutes. Then, with a discrete type of distribution, the entity would also require the use of a second resource (Resource_2) with an associated probability of 80%. The time for Resource_2 utilization would follow a Normal Distribution with mean 4 minutes and a standard deviation of 0,25 minutes.

There is a first-in first-out type of queue associated with each resource utilization.

The entity will then leave the system.

The Event Scheduling philosophy would identify three events – Entity Arrival Event (Figure 1); End of Resource_1 Utilization (Figure 2); End of Resource_2 Utilization (Figure 3).

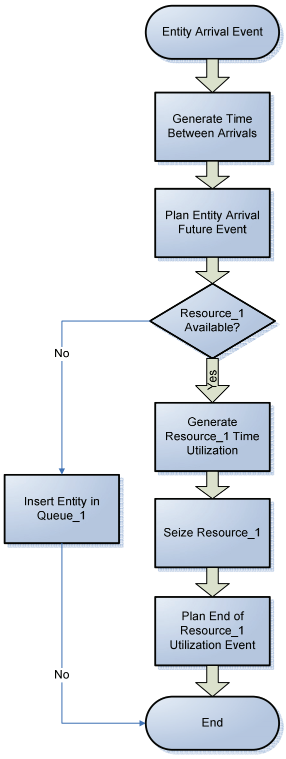

Entity Arrival Event
Generate Time Between Arrivals
Plan Entity Arrival Future Event
Resource_1 Available?
No
Yes
Generate Resource_1 Time Utilization
Insert Entity in Queue_1
Seize Resource_1
Plan End of Resource_1 Utilization Event
No
End

Figure 1 – Entity Arrival Event Flowchart

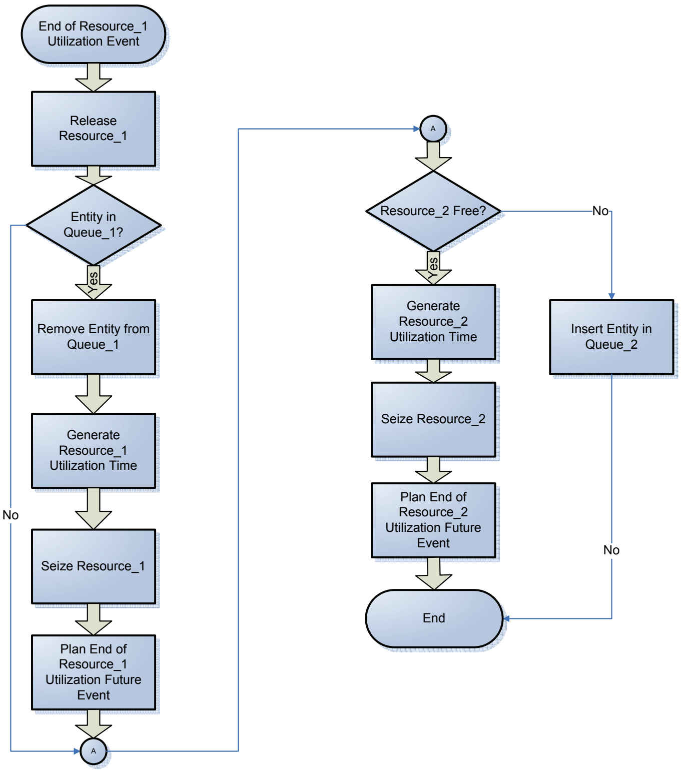

End of Resource_1 Utilization Event
Release Resource_1
Entity in Queue_1?
Yes
Remove Entity from Queue_1
Generate Resource_1 Utilization Time
No
Seize Resource_1
Plan End of Resource_1 Utilization Future Event
A
A
Resource_2 Free?
No
Yes
Generate Resource_2 Utilization Time
Insert Entity in Queue_2
Seize Resource_2
Plan End of Resource_2 Utilization Future Event
No
End

Figure 2 - End of Resource_1 Utilization Event Flowchart

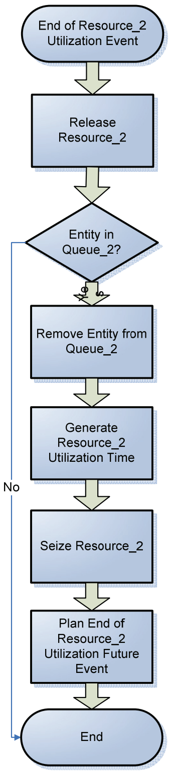

End of Resource_2 Utilization Event
Release Resource_2
Entity in Queue_2?
Yes
Remove Entity from Queue_2
Generate Resource_2 Utilization Time
No
Seize Resource_2
Plan End of Resource_2 Utilization Future Event
End

Figure 3 - End of Resource_2 Utilization Event Flowchart

The software tool would recognize these flowcharts upon the following (simplified) dialog:

```
Add Event
Event Name? Entity Arrival
Is it an Arrival Process? Y
Time Between Arrival? Exponential (3)
It will follow a Resource Utilization? Y
Which Resource? Resource_1
Probability associated? 100%

Add Event
Event name? End of Resource_1 Utilization
Is it an Arrival Process? N
Resource_1 Utilization Time? Random (5,9)
Number of Resource_1 Units Available? 3
It will follow a Resource Utilization? Y
Which resource? Resource_2
Probability associated? 80%

Add Event
Event name? End of Resource_2 Utilization
Is it an Arrival Process? N
Resource_2 Utilization Time? Normal (4, 0.25)
Number of Resource_2 Units Available? 2
It will follow a Resource Utilization? N
```

Following this dialog and according to the previous reference to the text file creation by the software tool, the text file below would be created:

```
Entity Arrival expo(3) 0 [Resource_1] [100]
Resource_1 Random (5,9) 3 [Resource_2] [80]
Resource_2 Normal (4,0.25) 2 [] []
```

As stated above and upon fully recognition of the characteristics of the text file constructed, the user could directly identify the text file corresponding to the flowcharts developed (skipping the dialog phase) and the software tool would automatically generate the Java Basic Simulation Facility program. This text file would also act as a starting point to future modifications of previous used examples.

This tool has been largely used in simulation classes, with informatics, information systems and industrial engineering students. They are asked to construct flowcharts for various examples and to use our tool. After a few initial enhancements, the tool has been generating correct Java programs. This has been the true verification procedure for this simulation tool.

4. CONCLUSION

The software tool presented shows three particularly interesting features:

- It is based on simple flowcharts that follow Event Scheduling Simulation philosophy
- It uses simple what-if questions to implement the respective flowcharts
- It automatically generates a Java computer program to perform the mimic of the system under analysis

These features, together, would contribute to

- the generalization and a better understanding of the use of simulation
- the automatic generation of simulation programs (*Java programs*)

The generalization and better understanding of the use of simulation would be accomplished once our tool only requires expertise in a basic simulation approach – Event Scheduling, thus incorporating simple flowcharts defining the system and its functioning rules.

Then, these simple flowcharts are translated into the software tool through a very simple questionnaire that is responsible for the automatic generation of a Java computer program that performs the mimic of the system and evaluates corresponding efficiency measures.

As far as future work is concerned, the authors would include in this tool the facility to deal with multiple queues examples. This feature would then recognize the ability for a resource to use entities from multiple queues, implementing both possibilities:

- when a resource uses entities from multiple queues
- when a resource uses an entity selected from a set of queues

REFERENCES

Clementson, A.T., 1982. *Extend Control and Simulation Language - ECSL*, Users Manual. Birmingham: CLE, COM Ltd.

Dias, L.S. and Rodrigues, A.G., 2002. Towards Simplicity in Modeling for Simulation. *Operational Research Society, Simulation Study Group, Two days Workshop*. March 20-21, Birmingham, UK.

Dias, L.S., Rodrigues, A.G. and Pereira, G.B., 2005. An Activity Oriented Visual Modeling Language with Automatic Translation to Different Paradigms. *Proceedings of the 19th European Conference On Modelling And Simulation (ECMS 2005)*, pp. 452-461. June 1-4, Riga (Latvia).

Dias, L.S., Pereira, G.B. and Rodrigues, A.G., 2007. A Shortlist of the Most Popular Discrete Simulation Tools. *Simulation News Europe*, 17 (1), 33-36.

Gilbreth F.B. and Gilbreth L.M., 1921. Process Charts - First Steps in Finding the One Best Way to do Work. *Presented at the Annual Meeting of The American Society of Mechanical Engineers*, New York, USA.

Nassi and Shneiderman, 1973. Flowchart techniques for structured programming, *ACM SIGPLAN* Notices, 12.

Pidd, M., 1992. *Computer Simulation in Management Science*. 3rd Ed. John Wiley & Sons, Inc.

Thesen, A., 1978. *Computer Methods in Operations Research*. New York: Academic Press.

Thesen, A. and Travis, L.E. 1990. Introduction to simulation (tutorial session). *Proceedings of the 22nd Conference on Winter Simulation*, pp. 14-21. December 09 – 12, New Orleans (Louisiana, United States,). O. Balci, Ed. IEEE Press, Piscataway, NJ.

Tocher, K.D., 1963. *The Art of Simulation*, UNIBOOKS – English Universities Press.

AUTHORS BIOGRAPHY

Luís S Dias was born in 1970 in Vila Nova de Foz Côa, Portugal. He graduated in Computer Science and Systems Engineering in the University of Minho, Portugal. He holds a PhD degree in Production and Systems Engineering from the University of Minho, Portugal. His main research interests are Operational Research, Simulation and Systems Visual Modeling.

Guilherme A B Pereira was born in 1961 in Porto, Portugal. He graduated in Industrial Engineering and Management in the University of Minho, Portugal. He holds an MSc degree in Operational Research and a PhD degree in Manufacturing and Mechanical Engineering from the University of Birmingham, UK. His main research interests are Operational Research and Simulation.

José A Oliveira was born 1966 in Matosinhos, Portugal. He studied Mechanical Engineering at the University of Porto, Portugal. He graduated with a Ph.D. in Production and Systems Engineering at University of Minho, Portugal. His main research interests are Optimization with Heuristic Methods in Systems Engineering.

HADA: TOWARDS A GENERIC TOOL FOR DATA ANALYSIS FOR HOSPITAL SIMULATIONS

I. Castilla[(a)], M. M. Gunal[(b)], M. Pidd[(c)], R. M. Aguilar[(d)]

[(a)(d)]Department of Systems Engineering and Automation, and Computer Architecture, La Laguna University, Spain
[(b)(c)]Lancaster University Management School, UK.

[(a)]ivan@isaatc.ull.es, [(b)]m.gunal@lancaster.ac.uk, [(c)]m.pidd@lancaster.ac.uk, [(d)]r.aguilar@ull.es

ABSTRACT

Discrete Event Simulation (DES) in healthcare modeling has been an active research area for many years. However one of the drawbacks of this method is the need for meaningful data for building valid models. This paper discusses Hospital Activity Data Analysis (HADA) which is software specifically designed to be used with a generic hospital simulation model (DGHPSim). The DGHPSim model is built for UK healthcare system and is presented conceptually in this paper. HADA integrates with raw data from different sources to evaluate a hospital's past performance. Its results can be used by hospital managers for statistical inference and general understanding, and by DGHPSim users for estimating appropriate parameters of the simulation models. As well as its use in DGHPSim HADA is well-suited to be generically used for any patient-flow type hospital simulation models.

Keywords: data analysis, discrete event simulation, healthcare

1. INTRODUCTION

It is a sad fact that many people must wait a long time before receiving the healthcare they need. The source of this problem in countries such as UK and Spain, where health care is financed through taxation, is the use of waiting lists in order to try to ration hospital care. The UK National Health System (NHS) has a long history of waiting lists as the service struggles to cope with a huge number of patients using limited resources.

Among other analytical techniques, Discrete Event Simulation (DES) has been widely used in health care analysis and improvement for many years. However, most DES applications tend to be highly focused with a microscopic scope on single services such as emergency departments (Jurishica 2005), outpatient clinics (Harper and Gamlin 2003), and operation theatre capacities (Sciomachen, Tanfani, and Testi 2005).

Although DES is known to be a flexible tool, and hence is used frequently in modelling in healthcare, one of its burdens in applications is the requirement for extensive data and its manipulation (Banks and Carson 1984). Data analysis is an important phase in the development of most simulation models. When dealing with a complex social system such as a hospital, some data may be easily obtainable but others may be very difficult to acquire, making it hard to obtain a clear representation of what the modeller wishes for (Jurishica 2005; Katsaliaki, Brailsford, Browning, and Knight 2005).

Modelling a hospital requires information (and data) from various sources such as a hospital information system, interviews with hospital staff, and personal observations at the hospital. All of these sources help the modeller gain understanding of the important aspects that need to be simulated.

Interviews and visits offer a qualitative view of the real system and interviews are challenging tasks in which the skills of the interviewer and the predisposition of the interviewee are crucial. Visits are useful if a general view of the hospital is required and can also fill some information gaps which can not be explained only with numerical data, such as the disposition of rooms and wards.

For large hospitals only source of verifiable, quantitative data from a hospital is its information systems from which the modeller may generate inputs and other characteristics of the model, though it is important to be wary of data generated by information systems when modelling (Pidd 2002). Moreover, being generally stored in databases, automated data extraction is possible. However, knowledge extraction is not as straightforward as it seems as the data structure used in the hospital information system may differ from those used in the simulation model. These differences can lead to heavy pre-processing and reorganization of hospital data. This problem is aggravated when trying to use the same simulation model for several hospitals: each hospital handles its own data structure and thus getting a generic transformation mechanism from data structure to simulation model is an extremely complex task. Other important sources of complexity include;

- incompatible data types (numeric data stored as text),

- different codifications for the same topic,
- missing data, and
- data errors

Although there are many generic simulation software packages and libraries which can be used to build a hospital simulation model, custom solutions for data analysis seem more appropriate for dealing with data problems. Certainly, a generic software which solves all the above problems is not achievable. However, a reasonable option is to make a specific software in a controlled environment that could be used to homogenize the data sources in order to reduce human interaction as much as possible.

This paper presents the development of a data analysis software which is specifically designed for estimating input parameters of a generic hospital simulation model (District General Hospital Performance Simulation-DGHPSim) [www.hospitalsimulation.info] built for evaluating hospitals' waiting time related performance in the UK. The model is not presented in detail but a discussion is given at conceptual level to provide enough information for a discussion of its data requirements.

2. A GENERIC WHOLE HOSPITAL SIMULATION MODEL

At conceptual level, a typical general hospital can be divided into three main parts: Accident and Emergency Department (A&E) for emergency patients, Outpatient Clinics for elective patients, and Inpatient wards/units for both elective and emergency patients. Patients arrive from the outside world and are, therefore endogenous. As well as entering via A&E, emergency General Practitioner (GP) referrals, are also significant and must be taken into account.

The DGHPSim suite comprises of four discrete event simulation models; A&E, outpatient, waiting list, and inpatient. These models are designed for simulating patient flows to a general hospital from a holistic view to investigate possible ways of reducing waiting times at various stages in patient journeys. The UK government's waiting time targets have put a great pressure on general hospitals in the UK, and hospital managements are forced to use their limited resources more efficiently than previously. The DGHPSim suite is generic and data-driven, that is, it can be fitted to particular hospitals by specifying parameters and other data.. A more comprehensive description of each sub-model can be found in Gunal and Pidd (2007b).

Not surprisingly Gunal and Pidd's main finding in building a generic hospital model is that although huge amounts of data are available with today's information systems in healthcare (hence it may seem like a heaven for simulation modellers), it is difficult to use these data to estimate system parameters which characterises a hospital. Hospital Activity Data Analyser (HADA) shown in Figure 1, is software designed to feed simulation models with the required inputs from the real world system. Note that the real data is not being used directly by the models but instead inferences from the data are used.

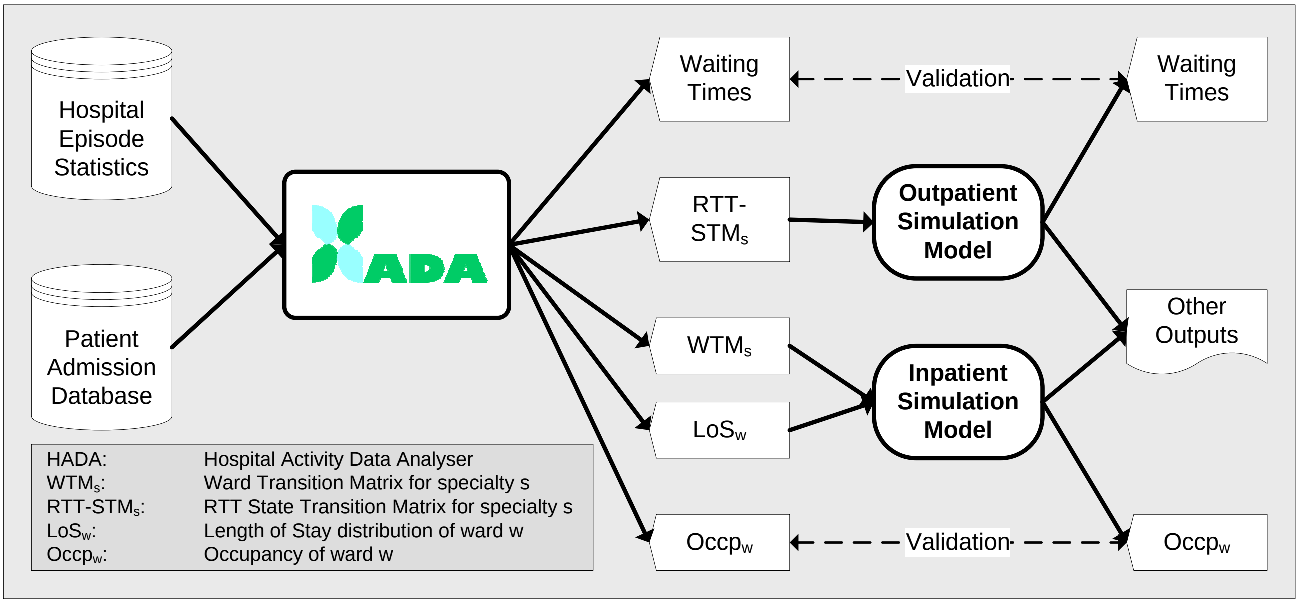

Hospital Episode Statistics
Patient Admission Database
HADA
Waiting Times
Validation
Waiting Times
RTT-STM$_s$
Outpatient Simulation Model
Other Outputs
WTM$_s$
LoS$_w$
Inpatient Simulation Model
Occp$_w$
Validation
Occp$_w$
HADA: Hospital Activity Data Analyser
WTM$_s$: Ward Transition Matrix for specialty s
RTT-STM$_s$: RTT State Transition Matrix for specialty s
LoS$_w$: Length of Stay distribution of ward w
Occp$_w$: Occupancy of ward w

Figure 1: Methodology for Estimating Input Parameters of Simulation Models

3. DATA SOURCES

There are two main sources of hospital data that are integrated in HADA: a hospital's patient admission system (PAS) and hospital episode statistics (HES).

3.1. Hospital Episode Statistics

HES is a UK-wide, routinely collected dataset capturing details of all inpatient and outpatient hospital episodes in the NHS. HES includes two huge datasets: one for outpatients (hereafter HESOP), which includes data regarding all outpatient appointments; and one for inpatients (hereafter HESIP), which gathers decision to admit, admission, operations. HESIP and HESOP structures are defined in HES-Online Data Dictionary (The NHS Information Centre 2008).

This data could be used to draw the general picture of Referral-To-Treatment (RTT) patient journeys of a hospital. However, there is no unique identifier which matches HESIP and HESOP to enable them to be linked to form patients' full journeys. Consequently, there is no way to link with absolute certainty episodes from outpatients and inpatients belonging to the same pathway.

3.2. Patient Admission System

Since HES is routinely collected for all hospitals, it can be seen as a source for generic simulation models. However, more detailed and customized models of a specific hospital would require a direct access to the hospital's information systems.

This data source, as opposed to HES data, is not homogeneous. Therefore, each hospital handles its own data and assumptions. Indeed, two different hospitals can even use the same term referring to different topics. For example, one hospital could treat spell and episode as synonyms.

Errors and incoherencies are also a frequent source of problems. In contrast to HES, which filters, or at least marks, most errors, hospital data can require a more comprehensive and careful review, and a more powerful error handling mechanism.

4. HADA

Hospital Activity Data Analyser (HADA) is standalone PC software which is designed for analyzing PAS and HES data for understanding a hospital's past performance as well as for estimating parameters of a hospital simulation model (such as DGHPSim). There is one software module per data source.

4.1. PAS Analysis

HADA can be used to evaluate hospitals' past performance based on the data provided to the software. The PAS data is pre-processed by HADA, with the help of a data conversion wizard, and processed to display information of two kinds: Bed occupancy, and LoS in each ward (or ward group).

This HADA's module relies on the identification of some basic fields in the original data source. The required fields are shown in Table 1.

Table 1: Fields Required from PAS Data

Field name	Meaning
Patient identifier	Anonymous patient Identification number
Spell number	Spell Identification number
Admission date	Admission date/time
Discharge date	Discharge date/time
Episode start date	Consultant (or bed) episode start date
Episode end date	Consultant episode end date
Elective date	For elective patients, decision to admit date
Ward code	Ward code number (or name)
Patient classification	Ordinary/Day case/Regular category
Admission source	Admission source
Admission method	Type of admission code (e.g. Emergency (21/22/23), Elective (11/12/13))
Primary diagnosis code	Diagnostic code
Consultant specialty	Consultant main specialty
Consultant	Anonymous consultant code

A user needs to identify these fields in the original data source. If the fields are not directly available, he or she should provide the software with a table where this data is accessible. The data conversion wizard allows the user to select a raw data source, to match the original fields with the corresponding expected fields, and, additionally, to add SQL statements which perform extra processing over the raw data. Once the wizard finishes, a preprocessed table is available. HADA makes use of this table to show a set of results.

By default, the resulting information about hospital wards and units is displayed by each ward. For example, if there are 20 wards in the PAS, the output can be displayed for each one of these 20 wards. Alternatively, wards (or units) can be logically grouped. For example if there are 4 general medicine wards in the hospital, they can be grouped as one, to be able to observe the general medicine wards' activity as whole. The grouping is especially useful and necessary for the transitions, which will be explained below.

HADA generates three kinds of outputs: bed occupancy, length of stay and transitions.

- HADA shows the number of occupied beds in two categories (Figure 2); Overnight stays (blue lines) and same-day discharges (red lines). This categorization is necessary mainly because HADA uses only “Admission Date” and “Discharge Date” to calculate occupancy. For example if a patient’s admission and discharge dates are the same, this means that the patient occupied one bed during the day however did not stay overnight. This is especially possible for observation wards (or units) such as Medical Assessment Wards, or Clinical Decision Units.
- The second type of output are related to the Length of Stay (LoS). Figure 3 shows the LoS histogram of the General Medicine wards from a UK hospital;. LoS histograms are especially useful to observe the LoS distributions in wards. One should generally expect some sort of decreasing curve (negative exponential), like the one in Figure 3.
- The PAS analysis section also generates the transition counts and probabilities of patients moving between wards whilst in the hospital. This is calculated for each specialty in the hospital, and for each type of patient (Emergency, Elective) separately. For example Table 2 shows the Ward Transition Matrix (WTM) for emergency patients admitted to general medicine. The first column shows “From” wards and the first row “To” wards. The “Gate” symbolizes the entrance and “Disc” the discharge. The table shows how 4711 patients are first admitted to the Assessment ward (Gate-GASM) and, of these, 2452 patients are transferred to a Medical ward (GASM-GMED); finally, 1712 patients are discharged from the Assessment ward (GASM-Disc). Remember that these wards could be actual wards in the hospital or the group of wards as the user defines; in this case, they are ward groups.

Use of WTMs in generic hospital bed management simulation models has been first introduced by Gunal and Pidd (2007a). This method depicts complex relationships between hospital units, based on historical data, and gives simulation model users full flexibility to experiment with different alternatives of bed configurations

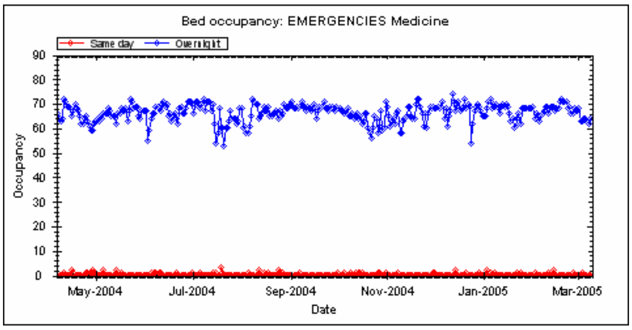


Figure 2: Bed Occupancy Related Outputs.

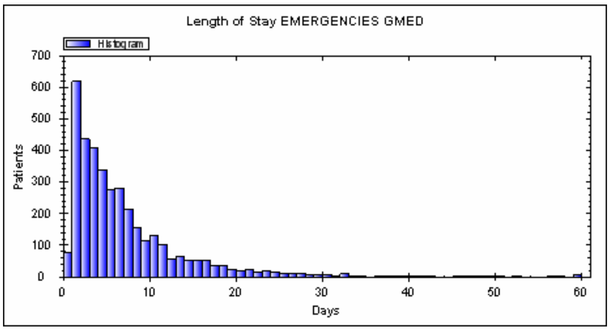


Figure 3: Length of Stay Distribution Related Outputs.

Table 2: General Medicine – Emergency Patients Ward Transition Matrix.

	GASM	GCAN	GCRI	GELD	GMED	GSPE	GSUR	GWOC	Disc
Gate	4711	10	455	9	397	3	15	3	0
GASM	24	125	80	242	2452	12	115	26	1712
GCAN	2	0	2	11	6	0	2	0	157
GCRI	25	3	8	5	203	0	14	0	330
GELD	6	0	4	50	19	0	9	0	472
GMED	18	41	32	224	15	23	159	64	2533
GSPE	0	0	0	2	5	0	2	0	30
GSUR	2	1	7	16	11	1	11	5	276
GWOC	0	0	0	1	1	0	3	0	93

4.2. HES Analysis

The second main function of HADA is related to analysis of the HES data. The final aim is to get a general picture of Referral-To-Treatment (RTT) patient journeys of a hospital (or a trust), that is, the succession of events which describe the different stages a patient goes through. This succession should include the events shown in Table 3.

Table 3: RTT Journey Events.

Event Type	Meaning	Source table
GP	First referral event, generally by a GP	HESOP
OP1	First outpatient appointment event	HESOP
OP2	Pre-operation follow-up outpatient appointment event	HESOP
IPDC	Inpatient as day-case admission event	HESIP
IPOR	Inpatient as ordinary admission event	HESIP
DADC	Decision to admit event (day-case)	HESIP
DAOR	Decision to admit event (ordinary)	HESIP
POP	Post-operation follow-up outpatient appointment event	HESOP
END	Discharged from consultant's care (in OP stage)	HESOP

As stated before, there is no implicit link between HESOP and HESIP tables apart from a patient identifier. It is not the objective of this analysis to detect relations between pathologies in a patient. Thus, it is assumed that different pathologies of the same patient are considered as different journeys. Consequently, we must rely on intuitive or heuristic knowledge in order to identify these journeys.

HES events consists of the following basic fields:

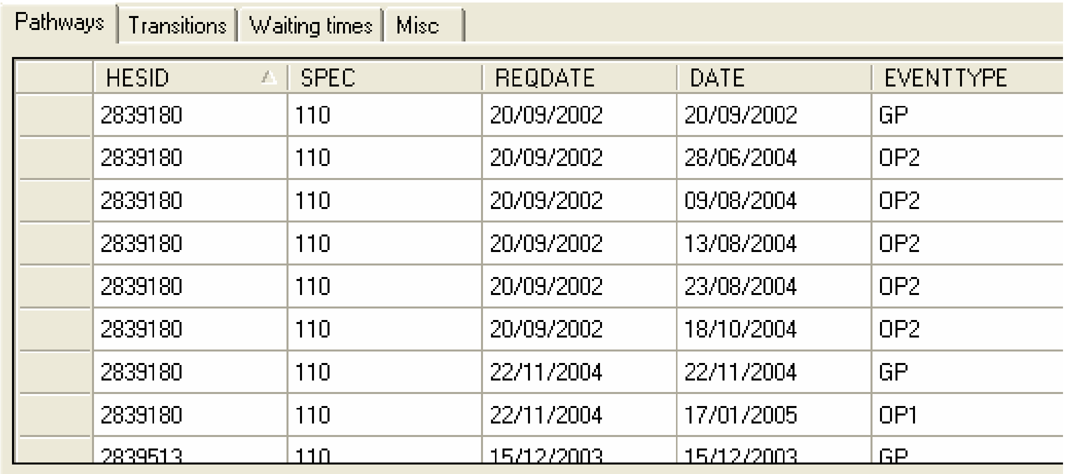
- HESID: Anonymized patient identifier in HES tables.
- Specialty: Specialty of the consultant who is responsible for the patient’s treatment.
- Referral Date: Referral date of the referrer (generally General Practitioners).
- Date: The date when this event happened.

HESID, Specialty and Referral Date constitute the patient journey key. Referral date is the trickiest part of this key, because it does not appear in HESIP.

HADA firstly identifies individual events in both tables. HESOP events are ordered by <HESID, Specialty, Referral Date>; and HESIP events are ordered by <HESID, Specialty, Admission Date, Episode End>. On the one hand, HESOP events with the same key constitutes a HESOP pathway, which is part of the whole patient journey; on the other hand, only pairs of (Decision to Admit, Admission) events can be identified as part of the whole patient journey in HESIP.

The next step is to link these pairs in HESIP with the corresponding HESOP pathway. A heuristic algorithm is used on this purpose. The algorithm looks for the first HESOP pathway with HESID and Specialty equals to one HESIP pair, and referral date previous to Decision to admit date. The Referral Date of this pathway is taken as the Referral Date of the whole patient journey, including HESIP events.

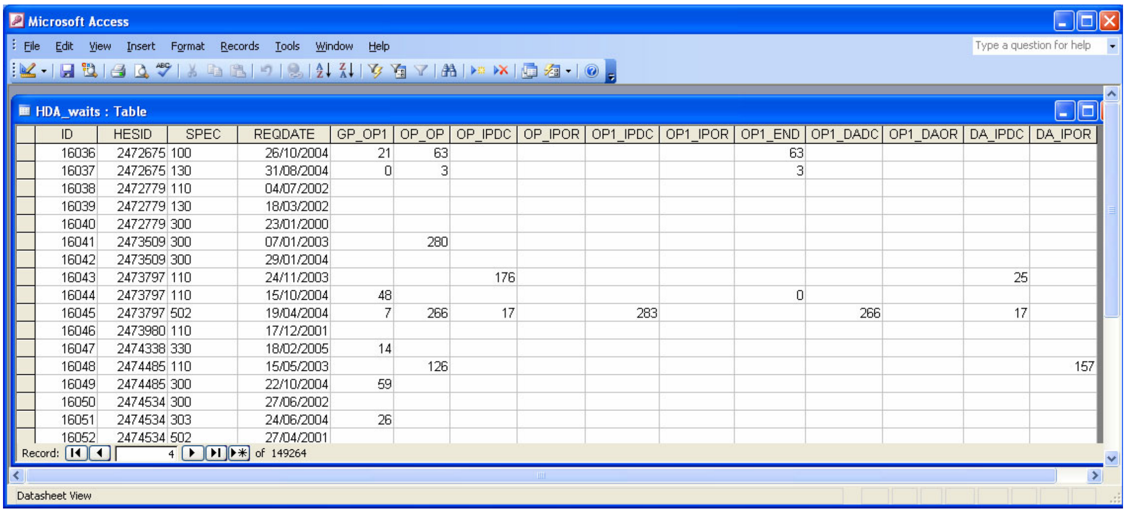
Once used, this HESOP pathway is disregarded and not used again. Thus, some HESIP pairs will not find their source referral date. These loose pairs are neglected. Figure 4 shows how HADA computes some patient pathways.



HESID	SPEC	REQDATE	DATE	EVENTTYPE
2839180	110	20/09/2002	20/09/2002	GP
2839180	110	20/09/2002	28/06/2004	OP2
2839180	110	20/09/2002	09/08/2004	OP2
2839180	110	20/09/2002	13/08/2004	OP2
2839180	110	20/09/2002	23/08/2004	OP2
2839180	110	20/09/2002	18/10/2004	OP2
2839180	110	22/11/2004	22/11/2004	GP
2839180	110	22/11/2004	17/01/2005	OP1
2839513	110	15/12/2003	15/12/2003	GP

Figure 4: Pathway Details Output Screen.

Based on the pathways table, HADA calculates the waiting times, or the delays between events. This is written to an MS Access table (Figure 5). Analyzing waiting times data is not done by HADA and is left to the users to do it externally, e.g. copying GP_OP1 column to Excel to draw a histogram.



ID	HESID	SPEC	REQDATE	GP_OP1	OP_OP	OP_IPDC	OP_IPOR	OP1_IPDC	OP1_IPOR	OP1_END	OP1_DADC	OP1_DAOR	DA_IPDC	DA_IPOR
16036	2472675	100	26/10/2004	21	63					63				
16037	2472675	130	31/08/2004	0	3					3				
16038	2472779	110	04/07/2002											
16039	2472779	130	18/03/2002											
16040	2472779	300	23/01/2000											
16041	2473509	300	07/01/2003		280									
16042	2473509	300	29/01/2004											
16043	2473797	110	24/11/2003			176							25	
16044	2473797	110	15/10/2004	48						0				
16045	2473797	502	19/04/2004	7	266	17		283			266		17	
16046	2473980	110	17/12/2001											
16047	2474338	330	18/02/2005	14										
16048	2474485	110	15/05/2003		126									157
16049	2474485	300	22/10/2004	59										
16050	2474534	300	27/06/2002											
16051	2474534	303	24/06/2004	26										
16052	2474534	502	27/04/2001											

Record: 4 of 149264

Datasheet View

Figure 5: Waiting Times Details Output Table in MS Access.

HADA's final output is the RTT State Transition Matrix. This is produced for every specialty and for all specialties separately. An example is given in Table 4. These matrices can be used for evaluating day-case and ordinary surgery rates, or follow-up and end of treatment percentages.

Table 4: General Surgery RTT State Transition Matrix for Specialty Code 100.

	GP	OP1	OP2	END	IPDC	IPOR	POP
GP	0	1	0	0	0	0	0
OP1	0	0	0.391	0.538	0.034	0.037	0
OP2	0	0	0.457	0.492	0.011	0.040	0
END	0	0	0	0	0	0	0
IPDC	0	0	0	0.601	0	0	0.399
IPOR	0	0	0	0.379	0	0	0.621
POP	0	0	0	0.487	0	0.105	0.408

5. CONCLUSIONS

We have discussed a generic software package for analyzing hospital activity data for a patient-flow hospital simulation model (DGHPSim). This software, HADA, can be used by analysts who wish to investigate a hospital's past performance, thus becoming an aiding software for better decision making, and for estimating input parameters for DGHPSim and other patient flow simulations. At this time, HADA has only been tested with DGHPSim, but future research includes tests with other simulation models.

HADA is intended to significantly reduce the time required by simulation users to set up input parameters. Hence, the use of this software makes patient-flow hospital simulation models more reusable.

ACKNOWLEDGMENTS

The DGHPSim project is funded by the Engineering and Physical Sciences Research Council of UK, under grant EP/C010752/1.

Iván Castilla is being supported by an FPU grant (ref. AP2005-2506) from the Ministerio de Educación y Ciencia of Spain.

REFERENCES

Banks, J. and Carson II, J.S., 1984. *Discrete-event system simulation*. Engelwood Cliffs, New Jersey: Prentice-Hall, inc.

Gunal, M.M. and Pidd, M., 2007a. Moving from Specific to Generic: Generic Modelling in Health Care. *Proceedings of the 2007 INFORMS Simulation*, July 5-7, INSEAD, France

Gunal, M.M. and Pidd, M., 2007b. Interconnected DES Models of Emergency, Outpatient, and Inpatient Departments of a Hospital. *Proceedings of the 2007 Winter Simulation Conference*, pp. 1461-1466. December 9-12, Washington, D.C. (USA).

Harper, P.R. and Gamlin, H.M., 2003. Reduced outpatient waiting times with improved appointment scheduling: a simulation modelling approach. *OR Spectrum*, 25(2): 207-222.

Jurishica, C.J., 2005. Emergency department simulations: medicine for building effective models. *Proceedings of the 2005 Winter Simulation Conference*, pp. 2674-2680. December 4-7, Orlando, FL (USA).

Pidd, M., 2002. *Tools for thinking: modeling in management science*. 2nd ed. Chichester: John Wiley & Sons Ltd.

Katsaliaki, K., Brailsford, S., Browning, D. and Knight, P., 2005. Mapping care pathways for the elderly. *Journal of Health Organization and Management*, 19(1): 57-72.

Sciomachen, A., Tanfani, E. and Testi, A., 2005. Simulation models for optimal schedules of operating theatres. *International Journal of Simulation: Systems, Science and Technology*, 6(12-13): 26-34.

The NHS Information Centre, 2008. *HES Online Data Dictionaries*. Available from: http://www.hesonline.org.uk/Ease/servlet/ContentServer?siteID=1937&categoryID=289 [Accessed 8 July 2008]

BIOGRAPHIES

IVÁN CASTILLA was born in La Laguna, Tenerife and went to the University of La Laguna, where he studied Engineering Computer Science and obtained his degree in 2004. He is currently developing his PhD with the department of Systems Engineering and Automation at the same university. His interests include discrete event simulation and business intelligence. Email to <ivan@isaatc.ull.es>.

MURAT GUNAL is pursuing his PhD at Lancaster University. He received his MSc degree from the same university in 2000. He is an experienced simulation modeller and his current research interest is to investigate the use of simulation on performance measurement in public services such as hospitals. Email to <m.gunal@lancaster.ac.uk>.

MIKE PIDD is Professor of Management Science at Lancaster University where his work spans simulation model-ling and the development of improved simulation methods utilising current developments in computing hardware and software. His current application work focuses on simulation modelling for improvement in public services, especially in policing and healthcare. He is known for three books: Computer simulation in management science (in its 5th edition), Tools for thinking: modelling in management science (in its 2nd edition) and Systems modelling: theory and practice – all published by John Wiley. Email to <m.pidd@lancaster.ac.uk>.

ROSA M. AGUILAR received her MS degree in Computer Science in 1993 from the University of Las Palmas de Gran Canaria and her PhD degree in Computer Science in 1998 from the University of La Laguna. She is associate professor in the Systems Engineering and Automation Department at the University of La Laguna, Canary Islands, Spain. Her current research interests are decision-making based on discrete event simulation systems and knowledge based systems, intelligent agents, and intelligent tutorial systems. Email to <r.aguilar@ull.es>

A MODEL TO DESCRIBE THE HOSPITAL DRUG DISTRIBUTION SYSTEM VIA FIRST ORDER HYBRID PETRI NETS

Mariagrazia Dotoli^(a1), Maria Pia Fanti^(a2), Agostino Marcello Mangini^(a3), Walter Ukovich^(b4)

^(a)Dipartimento di Elettrotecnica ed Elettronica, Polytechnic of Bari (Italy)
^(b) Dipartimento di Elettrotecnica, Elettronica e Informatica, University of Trieste (Italy)

^(1)dotoli@deemail.poliba.it, ^(2)fanti@deemail.poliba.it, ^(3)mangini@deemail.poliba.it, ^(4)walter.ukovich@deei.units.it

ABSTRACT

The paper proposes a model for simulation and performance evaluation of the hospital drug distribution system, a key process for the effectiveness and efficiency of the hospital offered services. In particular, we propose a modeling technique employing the first order hybrid Petri nets formalism, i.e., Petri nets making use of first order fluid approximation. The presented model is able to effectively describe the typical doctors and nurses daily operations and may be employed for staffing performance evaluation and optimization. A simulation of the drug distribution system of a department of an Italian hospital is performed in the well-known MATLAB environment to enlighten the potential of the proposed model.

Keywords: hospital drug distribution system, modeling, hybrid Petri nets, performance evaluation.

1. INTRODUCTION

The growing costs in the healthcare industry and the increasing demand for patient satisfaction force hospitals to improve their performance and effectiveness. The main tasks of a hospital can be synthesized in two objectives: i) *service quality* i.e., the effectiveness of the offered services; ii) *efficiency* as the reduction of the costs of drugs and staff, as well as the reduction of patient waiting times. The application of suitable management strategies in order to organize and coordinate the flow of people, drugs and information can be a key feature to reach such objectives of quality and efficiency.

In the related literature different approaches are investigated to improve the hospital processes and organization. For instance, Qi, G. Xu, Huo and X. Xu (2006) study the hospital management by applying industrial engineering strategies. Moreover, Kumar and Shim (2007) modify the business processes in an emergency department in order to minimize the patient waiting times. Furthermore, medical informatics systems and Internet related technologies are proposed and investigated in the literature (S.S. Choi, M.K. Choi, Song and Son 2005, Loh and Lee 2005). Usually, simulation is employed as a tool for verification and validation of the presented solution for the improvement of quality and efficiency. Typically, simulation is either carried out by way of a simulation software (Gunal and Pidd 2007, Kumar and Shim 2007), or by Petri Net (PN) models (S.S. Choi, M.K. Choi, Song and Son 2005, Xiong, Zhou and Manikopoulos 1994). However, the mentioned contributions in the related literature share the limitation that the solutions to improve the hospital management and organization are based on heuristic strategies.

This paper proposes a model to describe and optimize the drug distribution system in hospitals. Indeed, among the hospital processes and workflows, the drug prescription and distribution have a basic and key importance to improve service quality and efficiency (Taxis, Dean and Barber 1999). More precisely, the proposed model describes the drug distribution system starting from the prescription of medications by the doctor to the drug administering to the patient.

The presented model is based on First Order Hybrid Petri Nets (FOHPNs) (Balduzzi, Giua and Menga 2000) that are a hybrid PN formalism able to describe both the continuous and discrete dynamics of the system. Continuous places hold fluid, whereas discrete places contain a non-negative integer number of tokens and transitions, which are either discrete or continuous. FOHPNs present several key features. Fluid approximations provide an aggregated formulation to deal with complex systems, thus reducing the dimension of the state space so that the simulation can be efficiently performed. Moreover, the design parameters in fluid models are continuous; hence, there is the possibility of using gradient information to speed up optimization. In other words, the model allows us to define optimization problems of polynomial complexity in order to select suitable parameters that optimize appropriate performance indices.

The objectives of the proposed model are twofold. First, it describes the operations of doctors prescribing drugs and of nurses recording prescriptions and distributing drugs. Hence, the obtained model is suitable for describing the dynamics of drug distribution in hospital wards in order to analyze and simulate the system. Second, the fluid model allows us to optimize the number of doctors and of nurses that should be present in each work-shift to obtain satisfactory performance indices and service quality.

The obtained results are the starting study for the application to the hospital drug distribution system of innovative Information and Communication Technologies (ICT) tools as well as of recently developed electronics and informatics tools, in order to improve the system management.

The paper is organized as follows. Section 2 reports some basic definitions about the structure and dynamics of FOHPNs. Section 3 presents the description of the considered drug distribution system and the proposed FOHPN model. Moreover, Section 4 reports the simulation data and discusses the results obtained by the numerical simulation. Finally, Section 5 reports the conclusions.

2. FIRST ORDER HYBRID PETRI NETS

2.1. Net Structure and Marking

A FOHPN (Balduzzi, Giua and Menga 2000) is a bipartite digraph described by the six-tuple $PN=(P, T, \boldsymbol{Pre}, \boldsymbol{Post}, \Delta, F)$. The set of places $P=P_d \cup P_c$ is partitioned into a set of discrete places P_d (represented by circles) and a set of continuous places (represented by double circles). The set of transitions $T=T_d \cup T_c$ is partitioned into a set of discrete transitions T_d and a set of continuous transitions T_c (represented by double boxes). Moreover, the set of discrete transitions $T_d=T_I \cup T_S \cup T_D$ is partitioned into a set of immediate transitions T_I (represented by bars), a set of stochastic transitions T_S (represented by boxes) and a set of deterministic timed transitions T_D (represented by black boxes). We also denote $T_t=T_S \cup T_D$, indicating the set of timed transitions.

The matrices ***Pre*** and ***Post*** are the $|P| \times |T|$ pre-incidence and the post-incidence matrix, respectively. Note that $|A|$ denotes the cardinality of set A. Such matrices specify the net digraph arcs and are defined as follows: $\boldsymbol{Pre}, \boldsymbol{Post}: \begin{cases} P_c \times T \to \mathbb{R}^+ \\ P_d \times T \to \mathbb{N} \end{cases}$. We require that for all $t \in T_c$ and for all $p \in P_d$ it holds $\boldsymbol{Pre}(p,t)=\boldsymbol{Post}(p,t)$ (*well-formed nets*).

Function $\Delta : T_t \to \mathbb{R}^+$ specifies the timing of timed transitions. In particular, each $t_j \in T_S$ is associated with the average firing delay $\Delta(t_j)=\delta_j = 1/\lambda_j$, where λ_j is the average transition firing rate. Each $t_j \in T_D$ is associated with the constant firing delay $\Delta(t_j)=\delta_j$. Moreover, $F: T_c \to \mathbb{R}^+ \times \mathbb{R}_\infty^+$ specifies the firing speeds associated to continuous transitions, where $\mathbb{R}_\infty^+ = \mathbb{R}^+ \cup \{+\infty\}$. For any $t_j \in T_c$ we let $F(t_j)=(V_{mj}, V_{Mj})$, with $V_{mj} \le V_{Mj}$, where V_{mj} is the minimum firing speed and V_{Mj} the maximum firing speed of the continuous transition.

Given a FOHPN and a transition $t \in T$, we define sets $\bullet t=\{p \in P: Pre(p,t)>0\}$, $t\bullet=\{p \in P: Post(p,t)>0\}$ (pre-set and post-set of t). The corresponding restrictions to continuous or discrete places are ${}^{(d)}t=\bullet t \cap P_d$ or ${}^{(c)}t=\bullet t \cap P_c$. Similar notations may be used for pre-sets and post-sets of places. The net incidence matrix is $\boldsymbol{C}(p,t)=\boldsymbol{Post}(p,t)-\boldsymbol{Pre}(p,t)$. The restriction of $\boldsymbol{C}$ to P_X and T_X (with $X,Y \in \{c, d\}$) is $\boldsymbol{C}_{XY}$.

A marking $\mathbf{m}: \begin{cases} P_d \to \mathbb{N} \\ P_c \to \mathbb{R}^+ \end{cases}$ is a function assigning each discrete place a non-negative number of tokens (represented by black dots) and each continuous place a fluid volume; m_i denotes the marking of place p_i. The value of a marking at time τ is $\mathbf{m}(\tau)$.

The restrictions of $\mathbf{m}$ to P_d and P_c are $\mathbf{m}^d$ and $\mathbf{m}^c$. A FOHPN system $\langle PN,\mathbf{m}(\tau_0)\rangle$ is a FOHPN with initial marking $\mathbf{m}(\tau_0)$. The firings of continuous and discrete transitions are as follows: 1) a discrete transition $t \in T_d$ is enabled at $\mathbf{m}$ if for all $p_i \in \bullet t$, $m_i \ge \boldsymbol{Pre}(p_i,t)$; 2) a continuous transition $t \in T_c$ is enabled at $\mathbf{m}$ if for all $p_i \in {}^{(d)}t$, $m_i \ge \boldsymbol{Pre}(p_i,t)$. Moreover, an enabled transition $t \in T_c$ is said strongly enabled at $\mathbf{m}$ if for all $p_i \in {}^{(c)}t$, $m_i>0$; $t \in T_c$ is weakly enabled at $\mathbf{m}$ if for some $p_i \in {}^{(c)}t$, $m_i=0$. In addition, if $\langle PN,\mathbf{m}\rangle$ is a FOHPN system and $t_j \in T_c$, then its Instantaneous Firing Speed (IFS) is indicated by v_j and it holds: 1) if t_j is not enabled then $v_j=0$; 2) if t_j is strongly enabled, it may fire with any IFS $v_j \in [V_{mj}, V_{Mj}]$; 3) if t_j is weakly enabled, it may fire with any $v_j \in [V_{mj}, V_j]$, where $V_j \le V_{Mj}$ depends on the amount of fluid entering the empty input continuous place of t_j. In fact, t_j cannot remove more fluid from any empty input continuous place p^* than the quantity entered in p^* by other transitions.

We denote by $\boldsymbol{v}(\tau)=[v_1(\tau)\ v_2(\tau) \ldots\ v_{|Tc|}(\tau)]^{\mathrm{T}}$ the IFS vector at time τ. Any admissible IFS vector $\boldsymbol{v}$ at $\mathbf{m}$ is a feasible solution of the following linear set:

$$\begin{array}{ll} V_{Mj} - v_j \ge 0 & \forall t_j \in T_\varepsilon(\mathbf{m}) \\ v_j - V_{mj} \ge 0 & \forall t_j \in T_\varepsilon(\mathbf{m}) \\ v_j = 0 & \forall t_j \in T_\upsilon(\mathbf{m}) \\ \sum_{t_j \in T_\varepsilon} C(p,t_j) v_j \ge 0 & \forall p \in P_\varepsilon(\mathbf{m}) \end{array} \tag{1}$$

where $T_\varepsilon(\mathbf{m}) \subset T_c$ ($T_\upsilon(\mathbf{m}) \subset T_c$) is the subset of continuous transitions that are enabled (not enabled) at $\mathbf{m}$ and $P_\varepsilon(\mathbf{m}) = \{p_i \in P_c \mid m_i = 0\}$ is the subset of empty continuous places. The set of all solutions of (1) is denoted $S(PN,\mathbf{m})$.

2.2. Net Dynamics

The net dynamics combines time-driven and event-driven dynamics. *Macro events* occur when: i) a discrete transition fires or the enabling/disabling of a continuous transition takes place; ii) a continuous place becomes empty. The time-driven evolution of the marking of a place $p_i \in P_c$ is:

$$\dot{m}_i(\tau) = \sum_{t_j \in T_C} C(p_i, t_j) v_j(\tau) \,. \tag{2}$$

If τ_k and τ_{k+1} are the occurrence times of two subsequent macro-events, we assume that within the time interval $[\tau_k, \tau_{k+1})$ (macro period) the IFS vector $v(\tau_k)$ is constant. Then the continuous behavior of a FOHPN for $\tau \in [\tau_k, \tau_{k+1})$ is as follows:

$$\begin{aligned} \mathbf{m^c}(\tau) &= \mathbf{m^c}(\tau_k) + \boldsymbol{C}_{\mathbf{cc}} \mathbf{v}(\tau_k)(\tau - \tau_k) \\ \mathbf{m^d}(\tau) &= \mathbf{m^d}(\tau_k). \end{aligned} \tag{3}$$

The net evolution at the occurrence of a macro-event is:

$$\begin{aligned} \mathbf{m^c}(\tau_k) &= \mathbf{m^c}(\tau_k^{\,-}) + \boldsymbol{C}_{\mathbf{cd}} \boldsymbol{\sigma}(\tau_k) \\ \mathbf{m^d}(\tau_k) &= \mathbf{m^d}(\tau_k^{\,-}) + \boldsymbol{C}_{\mathbf{dd}} \boldsymbol{\sigma}(\tau_k), \end{aligned} \tag{4}$$

where $\boldsymbol{\sigma}(\tau_k)$ is the firing count vector associated to the firing of the discrete transition t_j.

3. DRUG DISTRIBUTION SYSTEM MODEL

3.1. System Description

The paper focuses on the drug distribution system of an Italian hospital department during each day of the week. Moreover, at this stage we assume that the work organization does not employ the modern ICT tools, a typical situation in most Italian hospitals.

The model input is represented by the patients that the doctors have to examine every day of the week. We assume that the number of patients is different each day. The system dynamics is as follows. Doctors visit patients daily and prescribe them drugs on a daily basis. Hence, the charge nurse records the prescribed medication on a book. Successively, nurses distribute and administer the different kinds of drugs. If the medications are stored in the department buffer storage, they are ready for administering. If this is not the case, medicines have to be ordered to the hospital pharmacy. Figure 1 shows a simplified scheme of the considered drug distribution system.

3.2. FOHPN Model

The FOHPN depicted in Figure 2 models the drug distribution to the patients present in the considered department each day of the week according to the scheme in Figure 1.

Every day, a certain number of patients is examined. The value of such an average number depends on the current day following a characteristic curve and is modelled by the weights b_j with j=1,…,7.

Accordingly, the stochastic transitions $t_j \in T_S$ with j=8,…,14 model the patient arrivals (we assume the hour as the time unit), while the timed discrete transitions $t_i \in T_D$, with i=1,…,7 and associated firing delay $\Delta(t_i) = \delta_i$ =24 h, represent the change of the day.

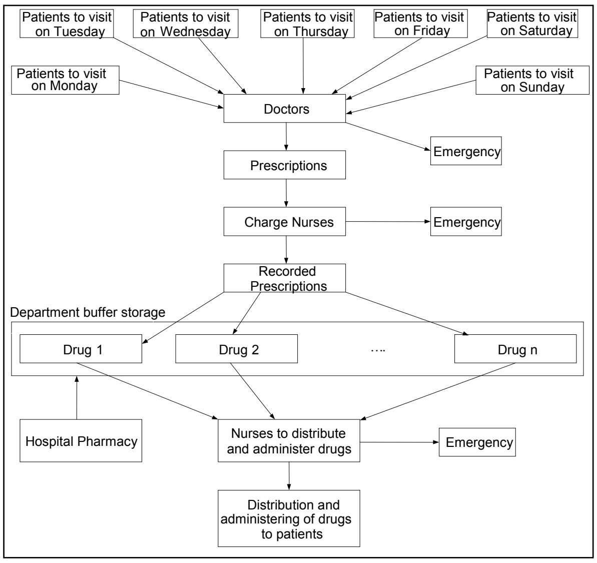

Patients to visit on Tuesday
Patients to visit on Wednesday
Patients to visit on Thursday
Patients to visit on Friday
Patients to visit on Saturday
Patients to visit on Monday
Patients to visit on Sunday
Doctors
Emergency
Prescriptions
Charge Nurses
Emergency
Recorded Prescriptions
Department buffer storage
Drug 1
Drug 2
....
Drug n
Hospital Pharmacy
Nurses to distribute and administer drugs
Emergency
Distribution and administering of drugs to patients

Figure 1: The scheme of the drug distribution system.

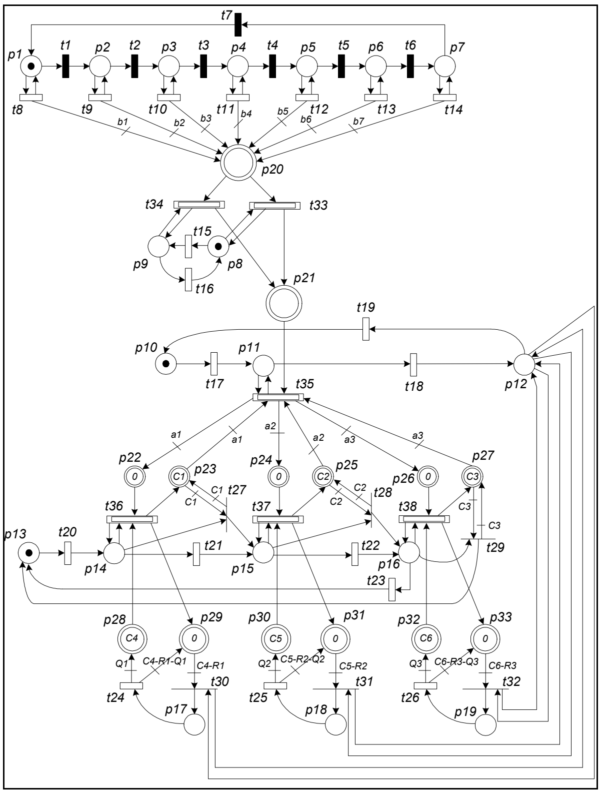


Figure 2: The FOHPN model of the considered drug distribution system.

Marking m_{20} represents the number of patients to be visited by the doctors. Place p_{20} enables two transitions t_{33}, $t_{34} \in T_c$ that model the works of the doctors in two alternative situations: t_{33} is enabled if the doctor has to do his rounds in the ward (p_8 is marked), whereas t_{34} is enabled if the doctor has to make an emergency visit (p_9 is marked). Consequently, the firing speed of transition t_{33} has to be assumed greater than that of t_{34}. After the patient examination, the doctor prescribes the drugs. Accordingly, the fluid content of the continuous place p_{21} represents the doctor prescriptions.

The charge nurse answers the emergency calls in the ward (place p_{10} marked), records the prescribed drugs on a book (place p_{11} marked and the corresponding transition t_{35} that represents the recording action enabled) and checks that the ward stockroom contains all the necessary drugs verifying the sell-by date (place p_{12} marked). Accordingly, the stochastic transitions t_{17}, t_{18} and t_{19} model the change of activity of the charge nurse.

The recorded prescriptions can correspond to different ways of administering medications. In Figure 2 we consider three main types of drugs administering: oral (place p_{22}), intravenous (place p_{24}), intramuscular (place p_{26}). Weights a_1, a_2 and a_3 represent the corresponding average number of oral, intravenous and intramuscular administerings present in a prescription.

When the charge nurse checks the ward stockroom, if the required drugs are not available or are inferior in number to a certain threshold R_i with $i=1,\ldots,3$ (the marking m_{29}, m_{31}, m_{33} exceeds the value C_4-R_1, C_5-R_2, C_6-R_3, with C_i with i=4,5,6 representing the capacity of the stockroom of oral, intravenous and intramuscular drugs, respectively), the charge nurse requests the medicines (t_{30}, t_{31}, t_{32}) to the hospital pharmacy. Hence, the workers provide the ward with the required drugs by the transitions t_{24}, t_{25} and t_{26} in the quantities Q_1, Q_2 and Q_3 for each type of medication, respectively. Note that the fictious capacities C_i with i=1,2,3 are included to avoid that the continuous transitions t_{36}, t_{37} and t_{38} are enabled even when no administering is required.

The last stage of the drug distribution system in the FOHPN model is represented by the nurses that administer the prescribed drugs following the indications of the record book (transitions t_{36}, t_{37}, t_{38}). Similar to the doctors and charge nurse, we consider also for the nurses the possibility that they have to satisfy an emergency call in the ward (place p_{13} marked).

Finally, the continuous places p_{23}, p_{25}, p_{27} represent the capacity of the medication buffers p_{22}, p_{24} and p_{26} and are necessary together with the instantaneous transitions t_{27}, t_{28} and t_{29} to shift to a different type of drug to administer if no prescription is present in the record book for the currently considered type of drug.

4. DRUG DISTRIBUTION SYSTEM SIMULATION

4.1. Simulation Specification

The dynamics of the FOHPN in Figure 2 is analyzed via simulation using the data in Tables 1 to 3 and the initial marking in the Figure. In the tables variables d, c and n represent respectively the number of doctors, charge nurses and nurses available in the department. The data of the simulated drug distribution system are obtained from interviews made to administration and medical staff of a typical department of an Italian hospital.

Table 1: Continuous transitions firing speed [hours] and description

Transition	$[V_{min}, V_{max}]$	Description
t_{33}	$[0, 3d]$	Patient examination under normal conditions
t_{34}	$[0, 2.1d]$	Patient examination under emergency
t_{35}	$[0, 12c]$	Prescription recording
t_{36}	$[0, 12n]$	Recording of administering of oral drug
t_{37}	$[0, 6n]$	Recording of administering of intravenous drug
t_{38}	$[0, 4n]$	Recording of administering of intramuscular drug

Table 2: Discrete transitions firing delay [hours] and description.

Transition	Average firing delay	Description
t_1, t_2, t_3, t_4, t_5, t_6, t_7	8	Work-shift duration on each day
t_8, t_9, t_{10}, t_{11}, t_{12}, t_{13}, t_{14}	1	Waiting time before examinations
t_{15}	7	Waiting time before an emergency
t_{16}	1	Time spent for emergency by a doctor
t_{17}	1	Time spent answering emergency calls by charge nurse
t_{18}	7	Time spent recording prescribed drugs by charge nurse
t_{19}	2	Time spent verifying stockroom levels by charge nurse
t_{20}	2	Time spent answering emergency calls by nurse
t_{21}	2	Time spent administering oral drug by nurse
t_{22}	2	Time spent administering intravenous drug by nurse
t_{23}	2	Time spent administering intramuscular drug by nurse
t_{24}	2	Time to replenish the inventory stocks of oral drug
t_{25}	2	Time to replenish the inventory stocks of intravenous drug
t_{26}	2	Time to replenish the inventory stocks of intramuscular drug
t_1, t_2, t_3, t_4, t_5, t_6, t_7	8	Work-shift duration on each day

Table 3: Weights, capacities, reorder levels and fixed order quantities [units].

a_1	a_2	a_3	b_1	b_2	b_3	b_4	b_5	b_6	b_7		
2	2	1	110	90	70	105	80	60	50		
C_1	C_2	C_3	C_4	C_5	C_6	Q_1	Q_2	Q_3	R_1	R_2	R_3
10^5	10^5	10^5	2000	1200	800	1400	800	600	20	20	20

In order to analyze the system behavior, the following performance indices are selected:

i) the average patient number visited per day;
ii) the average prescription number per day.

The simulations are performed in the MATLAB environment (The Mathworks 2006), an efficient software that allows us to model FOHPN systems with a large number of places and transitions. Such a matrix-based software appears particularly appropriate for simulating the dynamics of FOHPNs based on the matrix formulation of the marking update. Moreover, MATLAB is able to integrate modeling and simulation of hybrid systems with the execution of control and optimization algorithms.

We consider a simulation run of 112 time units, so that the total run time equals 2 weeks if we associate one time unit to one hour and 1 day to every 8 hours work-shift.

The simulation study is performed choosing the IFS vector of the continuous transitions $t_{33},\ldots,t_{38}$ in the FOHPN in Figure 2, representing the work rates of the doctors and nurses staff. At each macro-period the IFS vector $\boldsymbol{v}$ has to satisfy the constraint set (1) and has to maximize the sum of all flow rates:

$$\max\,(\mathbf{1}^{T}\cdot\boldsymbol{v}),\ \text{s.t. } \boldsymbol{v}\in S(PN,\boldsymbol{m}). \qquad (5)$$

In other words, such an operative condition allows us to estimate the maximum level of performance of the FOHPN system model, evaluated in terms of flows of patients, prescriptions and drugs to administer. By the knowledge of such rates it is possible to determine the optimal personnel dimension during each work-shift to obtain satisfactory performance indices and adequate service quality.

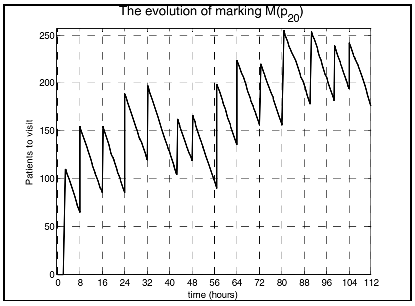


Figure 3: Patients waiting for examination (scenario 1: d=3, c=1, n=5).

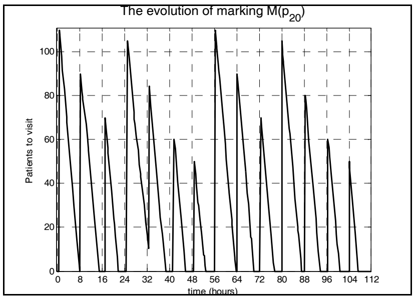


Figure 4: Patients waiting for examination (scenario 2: d=5, c=1, n=5).

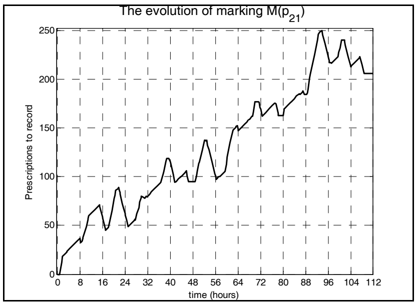


Figure 5: Prescriptions to record (scenario 2: d=5, c=1, n=5).

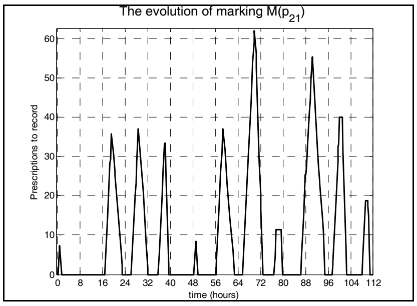


Figure 6: Prescriptions to record (scenario 3: d=5, c=2, n=5).

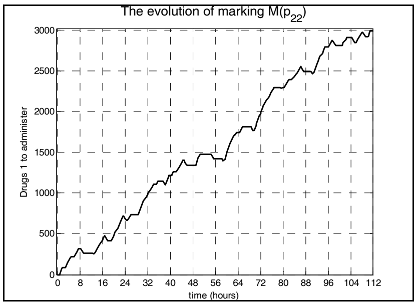


Figure 7: Oral administering to make (scenario 3: d=5, c=2, n=5).

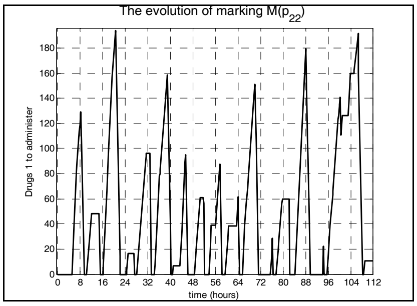


Figure 8: Oral administering to make (scenario 4: d=5, c=2, n=12).

4.2. Simulation Results

The simulations are performed for 4 scenarios, in which the staff dimensions d, c and n are varied.

In scenario 1 we assume d=3, c=1, n=5 and obtain a marking m_{20}, represented in Figure 3, that clearly corresponds to a situation in which the number of available doctors is insufficient to visit all patients. Indeed, the patient visits are not all performed in the established day.

In a second scenario, we increase *d*=5 and leave variables *c* and *n* unchanged with respect to scenario 1. Figure 4 shows that the number of patients to be yet examined at the end of each day (i.e., after each 8 hour work-shift) equals zero, i.e., every patient is visited in the predefined day. The only exception is registered at the end of the fourth day of the run time, when an emergency is present (note the variation of the rate of visits in the graph), but the patient surplus is coped with the next day. Hence, overall the number of doctors *d* is sufficient to deal with the average expected daily patient flow. However, marking m_{21}, represented in Figure 4, indicates that the only available charge nurse cannot record all prescriptions, so that the waiting prescriptions tend to increase indefinitely.

Hence, we consider a third scenario, with *d*=5, *c*=2 and *n* unchanged with respect to scenarios 1 and 2. Figure 5 visibly shows that the number of prescriptions still to be recorded at the end of each day is zero, i.e., overall the number of charge nurses *c* is now sufficient to cope with the average expected daily prescription flow. However, marking m_{22}, represented in Figure 7, shows that the available nurses are insufficient to administer the oral drugs (and the same can be shown to happen to the less frequently administered oral drugs), so that the waiting administerings tend to increase indefinitely.

A fourth and final scenario with *d*=5, *c*=2 and *n*=12 corresponds to a staff dimension such that all average daily flows of patients, prescriptions and administerings can be coped with by the department personnel. As an example, Figure 8 shows that with this configuration the (frequent) oral administering can be managed, except for an emergency that takes place in the second last day of the considered run-time and is coped with the next day.

5. CONCLUSION

To cope with the growing needs for service and quality in healthcare, we present a model for simulation and performance evaluation of the hospital drug distribution system. The proposed modeling technique employs the first order hybrid Petri nets formalism, i.e., Petri nets making use of first order fluid approximation. A simulation based on a hospital case study shows the effectiveness of the model in describing the system dynamics and in optimizing the hospital staff dimension to maximize the system performance, measured in terms of daily patients, prescriptions and drugs flows.

6. REFERENCES

Balduzzi, F., Giua, A., Menga, G., 2000. Modelling and Control with First Order Hybrid Petri Nets. *IEEE Transactions on Robotics and Automation*, 4 (16), 382-399.

Choi, S. S., Choi, M.K., Song, W.J., Son, S.H., 2005. Ubiquitous RFID Healthcare Systems Analysis on PhysioNet Grid Portal Services Using Petri Nets. *Proceedings of IEEE ICICS*, 1254-1258.

Gunal, M.M., Pidd, M., 2007. Interconnected DES Models of Emergency, Outpatient and Inpatient Departments of a Hospital. *Proceedings of the 2007 Winter Simulation Conference*, 1461- 1465.

Kumar, A., Shim S.J., 2007. Eliminating Emergency Department Wait by BPR Implementation. *Proceedings of the 2007 IEEE IEEM,* 1679-1683.

Loh, P.K.K., Lee, A., 2005. Medical Informatics System with Wireless Sensor Network-Enabled for Hospitals. *Proceeding of IEEE ISSNIP 2005,* 265-270.

Qi, E., Xu, G., Huo, Y, Xu, X., 2006. Study of Hospital Management Based on Hospitalization Process Improvement. *Proceedings of the IEEE IEEM,* 74-78.

The Mathworks, 2006. *MATLAB Release Notes for Release 14*. Natick, MA: The Mathworks.

Taxis, K., Dean, B., Barber, N., 1999. Hospital Drug Distribution Systems in the UK and Germany – a Study of Medication Errors. *Pharmacy World & Science*, 21 (1), 25-30.

Xiong, H.H., Zhou, M.C., Manikopoulos, C.N., 1994. Modeling and Performance Analysis of Medical Services Systems Using Petri Nets. *Proc. IEEE International Conference on Systems, Man and Cybernetics,* 2339-2342.

Mariagrazia Dotoli is since 1999 Assistant Professor in Systems and Control Engineering at the Department of Electrical and Electronic Engineering of Politecnico di Bari. Dr. Dotoli is the Associate Editor of the Mediterranean J. of Measurement and Control, of the Int. J. of Automation and Control and of the Int. J. of Systems Signal Control and Engineering Application.

Maria Pia Fanti is an Associate Professor of Systems and Control Engineering at the Department of Electrical and Electronic Engineering of Politecnico di Bari. Prof. Fanti is Associate Editor of IEEE Trans. on Systems, Man, and Cybernetics- Part A, IEEE Trans. on Automation Science and Engineering, the Mediterranean J of Measurement and Control, the Int. J. of Automation and Control, Enterprise Information Systems. She is Chair of the Central & South Italy Systems Man and Cybernetics IEEE Chapter.

Agostino Marcello Mangini is a Ph.D. student in Electrical Engineering at Politecnico di Bari and was a visiting scholar at the Zaragoza University. He was in the program committee of the 2007 IEEE International Conference on Systems, Man, and Cybernetics.

Walter Ukovich is Full Professor of Operations Research and director of the Department of Electrical Engineering, Electronics and Computer Science of the University of Trieste. His main research interests are in optimization, logistics and production planning and control. He has authored or co-authored over one hundred scientific papers on several scientific journals.

COMPLEX ORGANIZATION MODELING AND SIMULATION APPROACH FOR OPERATIONAL SCENARIOS STUDY: APPLICATION TO HEALTH CARE ORGANIZATION

A.S.Rebai, V.Chapurlat, D.Diep

LGI2P- Laboratoire de Génie Informatique et d'Ingénierie de Production
Site EERIE de L'Ecole des Mines d'Alès, Parc Scientifique Georges Besse
30035 Nîmes cedex 1 – France – tel. (+33) (0)466 387 066

ahmed-sami.rebai@ema.fr, vincent.chapurlat@ema.fr, daniel.diep@ema.fr

ABSTRACT

Any complex organization is confronted to hazardous or unpredictable events which are impossible to imagine and to anticipate due to the interactions and the nature of the components of the organization (person, machine, processes, etc.). Some risky situation may then impact the performance, the integrity and the stability of the organization. Actor involved into or in charge of this organization must then be able to detect and to characterize these events, the emerging situations and the possible induced risks. This paper presents a work on progress about a composite approach allowing to model a complex organization and to analyze its behavior when facing new situations. This approach is applied to Health Care Organizations.

Keywords: verification, validation, emergence, simulation, Multi Agents Systems

1. INTRODUCTION

Healthcare organizations are today facing to the same problematic from those which were considered in the last decade by industry. Indeed, they must improve their reactivity and flexibility in order to gain efficiency and to reach customer satisfaction. However, and it is a huge difference with industrial domain, they have in same time to respect simultaneously social, human, ethical and medical rules. Inherent complexity of such socio technical organizations (various actors, various disciplines, various technologies, etc.), medical environment constraints and objectives (various operational scenarios are to be achieved due to different medical pathologies, various situations, etc.) induces the occurrence of unpredictable events for which the organization reactions may be unsuitable and may cause prejudice to the patient. It seems interesting then to help managers to characterize this kind of event more precisely and to detect resulting situations i.e. emerging states in the behavior of the entire organization.

The research work on progress presented in this paper aims to integrate in an existing modeling and verification framework a simulation technique based on Multi Agents Systems. In the first part, the paper summarizes the problematic to be solved and present some existing works about simulation of emergence concept. The second part presents the used modeling and verification framework. The modeling technique is based on System Engineering (SE) (INCOSE 2004) and Enterprise Modeling (Vernadat 1996). The verification technique allows assuming model coherence is inspired by System Verification and Validation (V&V) (NASA 2001) and enterprise risk management (CAS 2003). The third part details the current work in progress which concerns the simulation technique based on multi agents system for helping model designers to make appear some relevant and sometimes emergent scenarios of evolution of the organization.

2. PROBLEMATIC AND APPROACH POSITIONNING

The strong interaction with a moving environment, the heterogeneity and the interactions between the components (services, human resources, machines, processes, etc.) of an organization induce the possible emergence (Chalmers 2002) of new behaviors and events. This concept of emergence is a multi-field concept which has resonances in biology, philosophy, artificial intelligence, etc. It characterizes the fact that an event occurring at a given level of detail cannot be deductible from properties of entities described at a more detailed level. Emergence taxonomy has been proposed by (Fromm 2005):

- Type I describes simple emergence without top-down feedback and self-organization, and includes intentional emergence.
- Type II contains the classic phenomena of weak emergence including top-down feedback and self-organization. It is further distinguished between stable and instable forms in this class.
- Type III covers all forms of emergence through multiple feedback and adaptation in more complex adaptive systems due to evolution.
- Type IV characterizes all forms of strong emergence in evolution. The term strong emergence is liberated from any magical or unscientific meaning.

This work focuses only on emergence of type I and

II. For example, a new behavior must emerge due to interaction between actors or a new scenario may be induced by a medical surgery. So how it is possible to detect emergent operational scenarios and situation of an organization composed of several resources working in parallel, wanting to reach their own objectives?

The most appropriate technology allowing simulating the parallel evolution of various complex entities independently from each other is based on Multi Agents Systems (MAS). A Multi-Agents System as a system made up of the following elements:

1. An environment E, i.e. a space generally having metric
2. A set of Objects O. These objects are located, i.e., for any object, it is possible, at a given moment, to associate a position in E. These objects are passive, i.e. they can be perceived, created, destroyed and modified by the agents.
3. A set A of agents, who are particular objects (A is included in O), which represent the active entities of the system.
4. A set of relations R who link objects (and the agents) between them.
5. A set of operations Op allowing the agents of A to perceive, to produce, to consume, to transform and manipulate objects of O.
6. Operations charged to represent the application of these operations and the reaction of the world to this attempt to modification, that we will call the laws of the universe.

Its main characteristics are (Wooldridge and Jennings 1995):

- located – the agent is able to act on its environment starting from the sensory perception which it receives from this same environment;
- autonomous – the agent is able to act without any intervention (human or agent) and controls its own actions as well as its internal state;
- proactive – the agent must produce a proactive and opportunist behavior, and being at the same time able to take the initiative at the good time;
- able to give a response in time – the agent must be able to perceive its environment and to elaborate a response in necessary time;
- social – the agent must be able to interact with other agents (software or human) in order to achieve tasks or to help these agents to achieve theirs.

There is then a consensus on the utility and the justification of the use of the multi-agents systems whose advantages are stated by (Brandolese, Brun, and Portioli-Staudacher 2000) to represent a complex system and to simulate efficiently complex interactions in behaviors:

- Dynamic system. The MAS inherits the IA symbolic treatment, i.e. knowledge. On the other hand, contrary to the traditional approaches of the Artificial Intelligence which simulate, to a certain manner, capacities of the human behavior, the MAS allow to model a set of agents which interact. The agents are structured in order to exert an influence on each one to make evolve the system in his totality (dynamic system). We find many interactions between agents such as coordination, negotiation, co-operation (Chaib-Draa, Jarras, and Moulin 2001). This approach is particularly well adapted to the simulation of the complex systems whose total operation emerges from the actions of the individuals. The MAS allow making virtually live autonomous agents on computer and to carry out there difficult experiments, even impossible to carry out in reality.
- A significant number of agents. A great number of agents is in the centre of the problem in this type of modeling contrary to the game theory where seldom more than three actors are represented.
- Flexibility of the data-processing tool which makes it possible to modify the behavior of the agents, to add or remove possible actions, to extend information available to the whole of the agents. The Multi-Agents model is made operational thanks to a data-processing implementation that does not impose any specific analytical requirement, but the use of the advanced data-processing languages: Object Oriented Language (OOL) who allows developing the program in a modular way. The processes programming at the local level in various modules and the use of individualized entities bring a great flexibility. The modifications do not require a broad reorganization of the program. A distributed resolution of problems. It is possible to break up a problem into under-parts and to solve each one independently to lead to a stable solution. This solution is not usually optimal within the meaning of complete rationality but it can be « satisfactory » within the meaning of Simon (Simon 1969).
- The MAS can "answer" to the individual failure of one of the elements, without degrading the system in its totality.

Thanks to these characteristics, the Multi-Agents Systems provide an approach for complex systems modeling and simulation appropriate to the study of emerging behaviors.

However, it is first necessary to provide a modeling framework of the organization independently from MAS architecture in order to permit actors who are not specialists from the field to describe the functions, the structure and the behavior of the organization (Aloui 2007). Second, it is necessary to check this model before rewriting it into a MAS system. This modeling framework includes verification mechanisms and is presented rapidly in the next part.

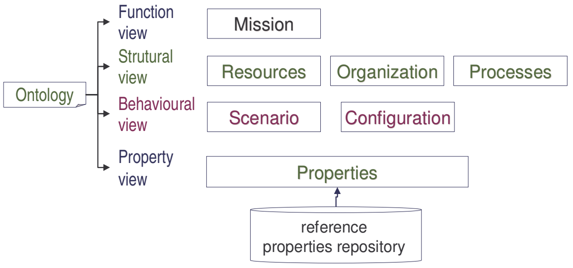
3. MODELING AND CHECKING FRAMEWORK

In order to better apprehend the complex socio technical system « healthcare organization», system engineering (SE) and Enterprise Modeling approaches are used. SE is defined according to (INCOSE 2004) by «*a co-operative and interdisciplinary approach for the progressive development and the checking of a solution for the system, balanced on the whole of its life cycle, satisfying customer expectations and acceptable by all* ». It is in particular deployed:

- To design, to make evolve and to check a system (hardware organized set, software, human competences and processes in interaction), bringing a solution to an operational need identified in accordance with measurable criteria of effectiveness.
- To satisfy requirements (quality, innovation, efficiency, delay, cost, performance, etc.) and constraints (reliability, safety, etc.) of the whole of its parts and being acceptable for the environment,
- To balance and optimize, under all the aspects, global economy of the solution on the whole life cycle of the system.

SE is based on systemic concepts and systemic reference model such as those proposed by the SAGACE approach (Penalva 1997). This approach provides particularly a grid modeling framework highlighting views. Each view gathers and formalizes the knowledge corresponding to a given aspect under which must be studied the pointed out system (here the health care organization). The resulting modeling framework is concerned by four views summarized in Figure 1. This figure shows the adapted modeling grid used in the following and the different modeling languages used in order to provide the models contained in each view. These views are:

- Functional: What is the mission of the organization? What is its finality i.e. why does it exists? How are we sure the organization provides the good mission with the appropriate level of performance? What are the functions the organization must fulfill in order to provide its mission?
- Behavioral: what are the possible evolution scenarios and configurations of the resources? How it evolves taking into account the environments and events? How it may be adapted and controlled in order to avoid damage in case of emergency?
- Structural: what are the processes which formalize the functional view of the organization? What are the current resources and their interaction in order to support these processes? How these resources are themselves organized during the organization life-cycle?



Functional view: *KAOS Methodology, IDEF 1*
Mission, finality, objectives
Structural view: *UEML, CEN 200*
Resources : Human, material, software
Organization : organizational units (service, department, etc.)
Processes
Behavioral view: *Synchronous Statechart, eFFBD (Enhanced Functional Flow Block Diagram)*
Scenarios, Configurations, Rules, adaptation and anticipation
Property view : *LUSP and Properties Reference Repository*
Modeling properties and identified risks

Figure 1: modeling framework synthesis

- Property: this view allows users to enrich the organization model by specifying properties the organization and its model must respect. A property (Chapurlat, Kamsu-Foguem, and Prunet 2003) expresses functional or non functional requirements (modeling requirement such as coherence rules of a model or semantic construction rules, organization requirement such as attribute evolution rules and limitation, expected behavior, functioning rule, constraint or objective), or potential risky situation. In this case, the concept of Cindynogenic Systemic Deficiencies (CSD) (Kervern 1994) has been used to model different kind of risks causes. A property is formally defined by a causal and typed relation linking two sets of events and data coming from each of the three over views of the organization model. Figure 2 shows an example of such property.

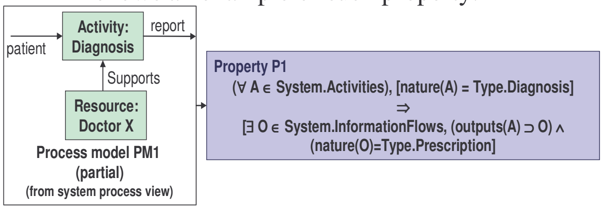

patient
Activity: Diagnosis
report
Supports
Resource: Doctor X
Process model PM1 (partial) (from system process view)
Property P1
(∀ A ∊ System.Activities), [nature(A) = Type.Diagnosis]
⇒
[∃ O ∊ System.InformationFlows, (outputs(A) ⊃ O) ∧
(nature(O)=Type.Prescription]

Figure 2: Property modeling a Cindynogenic Systemic Deficiency

This formal causal model allows checking the organization model coherence by using formal verification tools such as model checker or theorem prover.

Each of these four views may be expressed by different actors (modelers, engineers, specialists in the field to study here pharmacist, doctor, nurse, etc.)

involved into the organization. They have to explain and describe their own point of view thank to their own objectives. For this, a common and unique health care organization modeling ontology has been first defined. This one gathers commonly used and shared terms by all these actors for describing the main characteristics of the pointed out organization. In the same way, this ontology represents a unique, coherent and sufficient set of concepts and relations between concepts required for representing each view of the entire organization. In other terms, respecting the Model Driven Architecture (MDA) paradigm and avoiding interoperability problem between modeling languages, this ontology is built taking into account a unique and unified meta model. This one allows us to adapt and to unify some existing and pre selected modeling languages issued essentially from enterprise modeling and system engineering domains suitable to each view. For example, functional view uses the objective modeling language proposed by KAOS (Bertrand, Darimont, Delor, Massonet, and Van Lamsweerde 1998) and the IDEF-0 functional modeling language (Menzel and Mayer 1998). Unified Enterprise Modeling Language (UEML) (UEML 2003) allows describing resources organization, capabilities, processes and activity description and enhanced Functional Flow Block Diagrams (eFFBD) (Oliver, Kelliher, and Keegan 2004) permit to describe operational scenarios.

When the model is built, properties are checked on it in order first to assume its coherence, second to detect some potential causes of disturbances in the organization behavior.

If a property cannot be verified, the analysis process provides a counter example indicating the reasons for which the property is unsatisfied. Some properties allow detecting modeling errors or mistakes. These ones are used for checking the coherence of each view (coherence of the data and knowledge collected into a view and describe by using an unique modeling language dedicated to this view), and between each views (coherence of the data and knowledge collected and/or used in two separated views i.e. between models represented by using different modeling languages).

The proposed checking technique is based on a formal knowledge representation language called Conceptual Graphs (Sowa 1984) such as proposed in (Chapurlat and Aloui 2006). The technique is the following: The vocabulary used in the Conceptual Graph is formally extracted from the ontology. It is described with two lattices called respectively concepts lattices and relations lattices. Taking into account these lattices and a formal re writing algorithm presented in (Aloui 2007), the entire organization model i.e. all the models composing each view are translated into a unique conceptual graph. So this one gathers all the knowledge represented in the multi view and multi language modeling framework. This allows unifying the representation format of the required knowledge. Then it becomes possible to use mathematical foundations and associated mechanisms (Cogitant 2005). These ones allow to handle and to verify each property one by one and themselves translated under separated conceptual graphs. It remains however necessary to find a compromise between a completely formal verification, thus, exhaustive, and an ad hoc verification and validation technique. So simulation by using Multi Agents Systems has been chosen.

4. ANALYSIS FRAMEWORK: EMERGENCE AND MULTI AGENTS SYSTEMS

The following steps are proposed for simulating the behavior of a complex organization:

1. Rewrite the model in the form of an interacting network of agents. Cognitive agents are specifically developed in order to describe resources and processes behavior. For this, a set of mapping rules have been proposed to ensure the rewriting step of the Healthcare organizational model into Multi Agents System. Figure 3 shows this step.

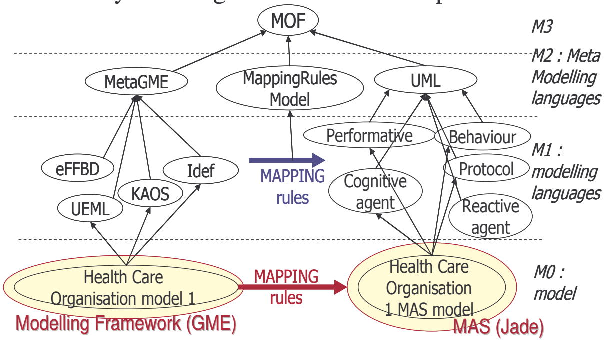


Figure 3: rewriting step

   Models described in the behavioral view are translated in the form of agents thanks to the information gathered into the functional and organizational views and according to the partial meta model shown in figure 4.

2. Some operational scenarios have been described such as those proposed in Figure 5. The goal is to simulate evolution of resources and processes all along this scenario and to make appear possible divergence between the proposed scenario and the scenario proposed by the MAS system. This requires the observation of the evolution of the different agents.
3. On each evolution 'step', try to prove properties with an impact on the behavior of the agents: properties on the interactions between agents, on the behavior of each agent considered separately, etc. This allows to modify and to adapt the behavior of each cognitive agent. Indeed each property may induce new events and the required psychological evolution of the modelled resources. So, during the next evolution step, the behavior of the organization may evolve from an unpredictable manner.

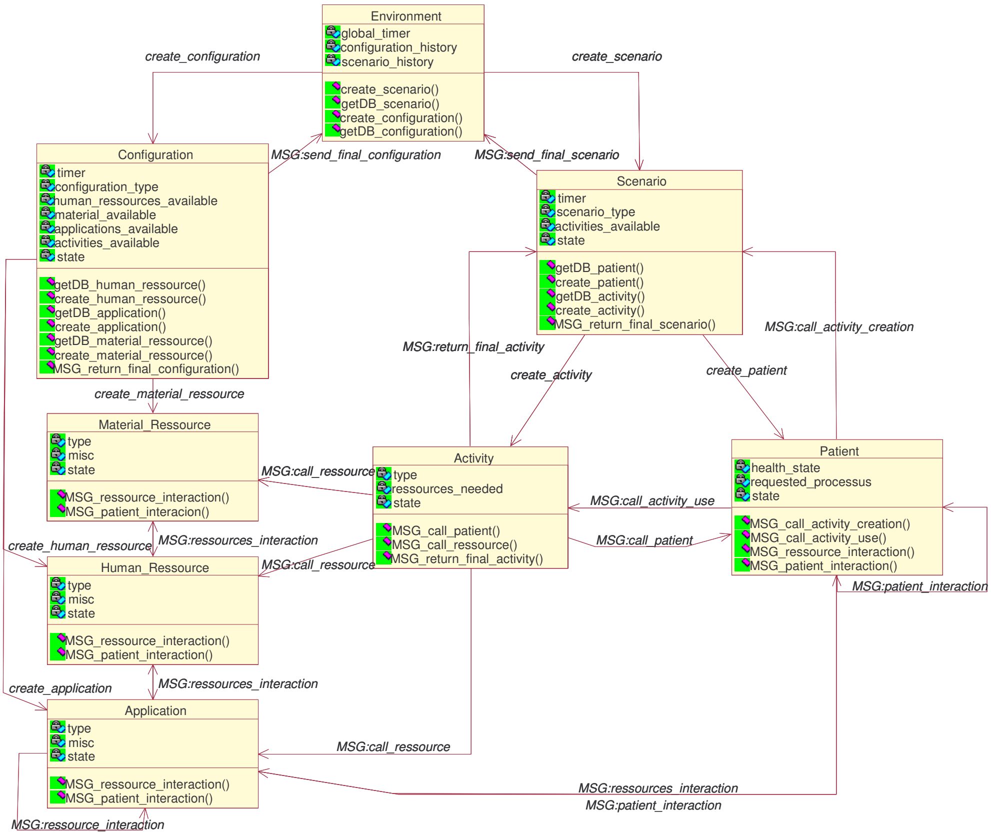


Figure 4: Meta model (partial view)

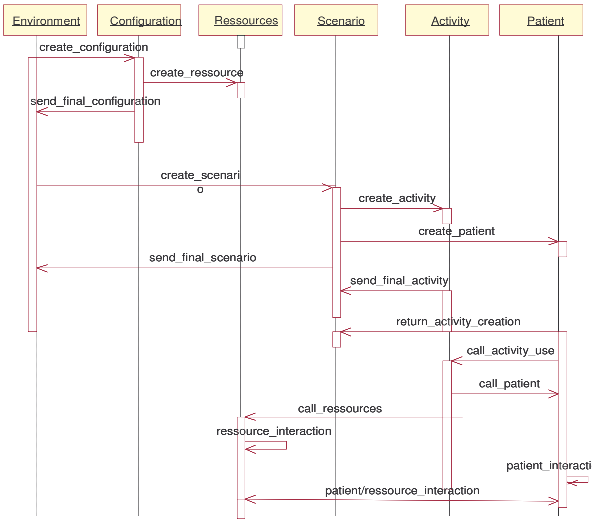


Figure 5: scenario example

4. The idea of this step is to improve the model by executing it. By considering a set of simulations results, the original organization model may be improved by taking into account the new potential scenarios.

A new cycle of modeling phase, coherence analysis phase by properties proof and simulation phase can start as summarized in Figure 6. For this, many MAS architecture have been developed and the one witch interest us is based on Believe, Desire and Intention paradigm (BDI) (Bratman, Israel, and Pollack 1988; Rao and Georgeff 1995). This architecture allows the description of the behavior of each agent as a set of learning rules and behavioral constraints inspired by the properties to be verified all along the life cycle of the organization. During the simulation, each agent must be then able to prove these properties and eventually may modify and adapt its behavior. That’s why, in BDI profile, we propose to put formalized properties in the form of rule and to adopt a JESS mechanism of reasoning as proposed in (Cardoso 2007). JESS can be used as a decision component of an agent, which is implemented in a declarative way.

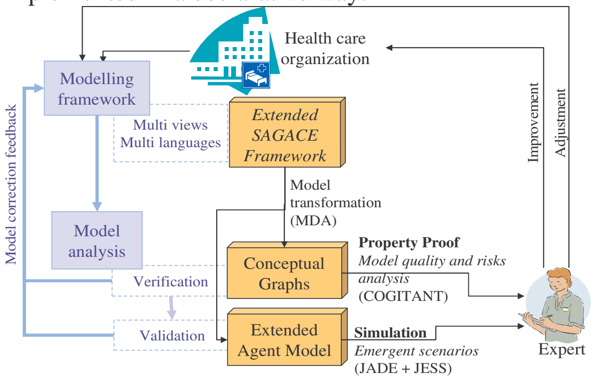


Figure 6: Approach synthesis and resulting framework

The proposed implementation consists of embedding an instance of the JESS engine inside the agent behavior. Since the agent must remain able to continuously reason, a formal proof mechanism is added whose action consists of running the Jess engine and verify in each time the set of required properties. In other words, partial and local dynamic evolution rules (described as a temporal property) can interact on the local agent behavior modifying then the resulting global behavior when one cannot precisely describe this global behavior.

So, each agent is equipped with a clean system of local proof properties (trade, normative and interoperability) that makes it possible on each evolution of its environment to test its current configuration, then to decide and/or improve, to extend or reduce this configuration, impacting its behavior. The goal is to facilitate a phenomenon of adaptation even of auto-organization. But this behavioral evolution of each agent is not possible all the time and evolution situation is then blocked due to the difficulty for agent to adapt their behavior. So organization is facing a new emergent risky situation having to be studied.

5. CONCLUSION AND FUTURE WORKS

In this paper, we have summarized existing results concerning health care organization modeling and analysis frameworks, verification technique developed for assuming model coherence. The current work in progress is presented. It concerns the dynamic validation of property based on BDI architecture of MAS using JESS engine as a mechanism of formal proof.

REFERENCES

Aloui, S., 2007. System engineering and enterprise modeling for risks management: application to the drug circuit in a university hospital. *URMPM - Union of Risk Management for Preventive Medicine 2nd American Congress - Improving the quality and sustainability of health care services*, 14-15 june, Montreal, Canada.

Bertrand, P., Darimont, R., Delor, E., Massonet, P.,Van Lamsweerde, A., 1998. GRAIL/KAOS: an environment for goal driven requirements engineering, *20th International Conference on Software Engineering, IEEE-ACM*, april, Kyoto.

Brandolese, A., Brun, A., Portioli-Staudacher, A., 2000. A Multi-Agent approach for the capacity allocation problem. *International Journal of Production Economics* Vol. 66: pp. 269-285.

Bratman, M. E., Israel, D., and Pollack, M. E., 1988. Plans and resource-bounded practical reasoning. *Computational Intelligence* Vol. 4: pp. 349-355.

Cardoso, H.L., 2007. *Integrating JADE and JESS*, Available from: http://jade.tilab.com/doc/tutorials/jade-jess/jade_jess.html.

CAS, 2003. Casualty Actuarial Society (CAS), *Overview on Enterprise Risk Management*. ERM Committee.

Chaib-Draa, B., Jarras, I., Moulin, B., 2001. *Systèmes multi-agents : principes généraux et applications.* In Briot J.-P. and Demazeau Y. Eds. *Principes et architectures des systèmes multi-agents*. Hermès.

Chalmers, D. J., 2002. *Varieties of Emergence.* Available from: http://consc.net/papers/granada.html.

Chapurlat, V., Kamsu-Foguem, B., Prunet, F., 2003. Enterprise model verification and validation: an approach. *Annual Review in Control* Vol. 27, Issue 2: pp 185-197.

Chapurlat, V., Aloui, S., 2006. How to detect risks with a formal approach? From property specification to risk emergence. *Proceedings of Modelling, Simulation, Verification and Validation of Enterprise Information Systems (MSVVEIS)*, Paphos, Cyprus.

Cogitant, 2005. *CoGITaNT Version 5.1.* Reference Manual Available from: http://cogitant.sourceforge.net.

Ferber j., 1995. *Les systèmes multi-agents : Vers une intelligence collective*. Editions InterEditions.

Fromm, J., 2005. *Types and Forms of Emergence. Kassel University Press*.

INCOSE, 2004. *A «How To» Guide For All Engineers*. Working Group, System Engineering Handbook. INCOSE.

Kamsu-foguem, B., and al., 2003. Entreprise model verification: a graph based approach. *CESA'03, Symposium on discrete event in industrial and manufacturing systems; IEEE/SMC multiconference on computational engineering in systems application*. July, Lille France.

Kervern, G.Y., 1994. *Latest Advances in Cindynics*. Economica.

Menzel, C.P., Mayer, R.J., 1998. *The IDEF Family of Languages.* In Bernus P., Mertins K., and Schmidt G. Eds. *Handbook on architectures of information systems*. Berlin, Springer.

NASA, 2001. *VV&A Recommended Practices Guide: Glossary*.

Oliver, D.W., Kelliher, T.P., Keegan, J.G., 2004. *Engineering complex systems with Models and Objects*. McGraw-Hill.

Penalva, J.-M., 1997. *La modélisation par les sytèmes en situations complexes.* PhD Thesis, Paris Sud University.

Rao, A., Georgeff, M., 1995. BDI Agents: From Theory to Practice. *Proceedings of the First International Conference on Multi-Agent Systems (ICMAS-95).* AAAI Press, pp. 312-319, Menlo Park, California.

Simon, H. E., 1969. *The Sciences of the Artificial*. Cambridge: MIT Press.

Sowa, J.F., 1984. *Conceptual structures: information processing in mind and machine*. New York (U.S.A.): Addison-Wesley.

UEML, 2003. *Deliverable D3.1: Requirements analysis: initial core constructs and architecture, Unified Enterprise Modelling Language UEML Thematic Network - IST-2001-34229*. Available from: www.ueml.org.

Vernadat, F.B., 1996. *Enterprise Modelling and Integration: Principles and Applications*. Chapmann & Hall.

Wooldridge, M., Jennings, N., 1995. Intelligent Agents: Theory and Practice. *Knowledge Engineering Review*.

MOBILE GAME-BASED LEARNING SIMULATIONS FOR CRISIS MANAGEMENT TRAINING. A CASE STUDY “MOGABAL FOR E-HEALTH”

Mininel S[a], Vatta F[a], Gaion S[a], Ukovich W[a]

[a]University of Trieste
[a]mininel@gnbts.units.it

ABSTRACT

The mobile games sector is an important growth area for the games industry and research indicates the potential of mobile games to encourage learning in young adults. The 3-year EC-supported project mGBL (mobile Game-Based Learning) has the objective to prototype a platform for the development and deployment of mobile learning and guidance games, able to support the learning process and the support of decision making in critical situations not only in a cognitive but also in an emotional way. This paper describes key issues emerged in development phase of the Mogabal game engine within mGBL framework in both technical and pedagogical aspects, showing technologies, strategies and methodology adopted. We present the game prototype developed for mGBL 2nd User Trials with “triage” and “first aid” contents as a case study showing potentialities and development possibilities in the field of crisis management training.

Keywords: game-based learning, mobile devices, simulation, crisis management training

1. INTRODUCTION: VIDEOGAMES AND GAME-BASED LEARNING

Videogames industry has been growing at impressive rate in the last years, even succeeding, according to several reports, in historical surpasses collecting an early revenue superior to those of film and music industries respectively. While there are well known issues on part of some associations regarding violent contents, it is a generally accepted evaluation that many successful mainstream videogames contained interesting educational contents, although sometimes that happened just as a side-effect of game mechanics.

There is a consistent effort, in literature and in industry, to exploit the potentialities of videogames in a more focused learning effort: the so called Game-Based Learning. Not only does the integration of learning with gaming make science more fun; it also:

- Motivates students to learn
- Immerses them in the material so they learn more effectively
- Encourages them to learn from their mistakes

A wide body of research documents the pedagogical role of fun in learning (Fabricatore 2000). For example strategic use of games in learning programmes can contribute a ‘flow’ experience that is a characteristic of successful learning processes.

S. Johnson argues that video games in fact demand far more from a player than traditional games like Monopoly. To experience the game, the player must first determine the objectives, as well as how to complete them. They must then learn the game controls and how the human-machine interface works. Beyond such skills video games are based upon the player navigating (and eventually mastering) a highly complex system with many variables. This requires a strong analytical ability, as well as flexibility and adaptability. He argues that the process of learning the boundaries, goals, and controls of a given game is often a highly demanding one that calls on many different areas of cognitive function.

An important and emerging aspect in videogames industry is the one related to the so-called mobile gaming, that is, gaming on personal and portable mobile technologies: mobile phones, portable gaming devices, Personal Digital Assistants (PDA’s). Initially game use on this kind of devices was severely limited by technology issues (limited capabilities in computing power, screen resolution, etc. ...) but quickly evolving hardware allows development of more and more complex and interesting games for such devices.

2. MOBILE GAME-BASED LEARNING AND THE MGBL PROJECT

A growing body of research (Bellotti 2003; Ha 2007) indicates that mobile technologies can be effective tools in catering for students in a digital age and there are signs of the motivating potential and possible learning gains of games played on mobile devices with young adult audiences. With the growing sophistication and affordability of mobile technologies and applications, their use as learning tools becomes increasingly viable, hence the growing interest in the field of m-learning. However, in seeking to cater for the learning needs of young audiences who in general have high relation to mobile technologies, merely trying to adapt e-learning approaches for use with mobile technologies will not be enough. Young adults in particular need m-learning opportunities that are not only cognitively accessible but that also engage them in affective learning.

mGBL is a 3-year research and development project that is supported by the European Commission (EC) within the Information Society Technologies (IST) programme in the Sixth Framework. The project began in October 2005 and at the time of writing is in its third year. Ten partner organizations from EC countries Austria, Italy, Slovenia and UK and from associate country Croatia form the consortium, which is led by evolaris Privatstiftung in Graz, Austria. Target audiences are young adults aged 18 – 24, who regularly use mobile technologies. Project's challenge is to design exciting learning games that young adults will find fun to use. The focus is on supporting the development of decision-making skills for use in critical situations, a key area of concern in the EC. Three game models have been developed within the project, with example contents in the fields of e-commerce, e-health and career guidance, which are areas of strength within the consortium.

2.1. Pedagogical Framework

Specific theories informing the mGBL game design process include theory of the 'zone of proximal development' (Vygotsky 1978), i.e. the level of development that learners achieve when they engage in social behaviour is greater than they can achieve when working alone. We also base on experiential learning theory (Kolb 1984), with particular focus on learning phases: Planning, Doing, Feedback, Digesting (Race 1994). To cater for different learner preferences, we use the Theory of Multiple Intelligences (Gardner 1983), which proposes that human intelligence is a mixture of several intelligences. Finally, to support the development of decision-making skills and creative problem-solving strategies, we make use of the concept of 'single loop' and 'double loop' reflection (Argyris 1976), understood here as follows:

- Co-operation (following agreed rules, procedures and reflecting on efficiencies achieved thereby - 'single loop reflection');
- Collaboration (open sharing of ideas - challenging basic assumptions - 'double loop reflection').

This conceptual framework is seen to fit the benefits of mobile technologies, with their potential for making learning opportunities available anywhere, anytime and for social interaction and collaborative learning.

3. MOGABAL GAME ENGINE: MATERIALS AND METHODS

The mGBL game developed by the University of Trieste has been named "Mogabal" (extending the project acronym of MObile GAme-BAsed Learning). It is more correct to define it as a sort of game engine rather than a game, as its graphical aspects, rules, educational contents and many other elements can be fully configured and altered thus giving the potentiality for creation of widely different games and game styles.

The Mogabal game engine was developed in Java ME and is then playable on any mobile device having a Java Virtual Machine compliant with Java Micro Edition (J2ME) (Lei, Hui 2006), CLDC 1.0 and MIDP 1.1.

Java development environments were Eclipse and NetBeans, with use of the J2ME Polish 2.0 suite of tools to manage portability of code towards different mobile devices and language localization.

4. MOGABAL GAME ENGINE: TECHNICAL SOLUTIONS

The challenge, from the technical point of view, in programming a J2ME game in the mGBL framework, is that the objective is not to create "one" game, but the potentiality for wide range of games. Key issues in design were as follow:

- the game type and style may be highly variable;
- the game pedagogical content is highly variable;
- both game type and its pedagogical content are to be created, modified or customized by users (the teachers creating the game as support to their courses) in a relatively easy way.

In fact, the objective was to create something that could be modified with user-friendly tools without need of knowledge of Java programming, beyond some basic knowledge of steps necessary to modify the Java archive file to be installed on the mobile device.

The solution, regarding the game mechanics, was found in the use of the relatively simple game concept of movement on a rectangular map, having different graphical layers: background graphics, active elements ("sprites") and possible "fog of war".

The player's avatar roams the map. If "fog of war" is present, initially the map is obscured and revealed only during map exploration. The collision with one of the sprites launches an "event". Each sprite is tagged with a three characters code and a list of "1 to N" events is associated with that code. According to game construction rules, sprites can be programmed to "disappear" from the game after being collided, or they can be "permanent" and launch a random event taken from the list of events with the same code at each collision with players' avatars.

The different kinds of "events" supported are:

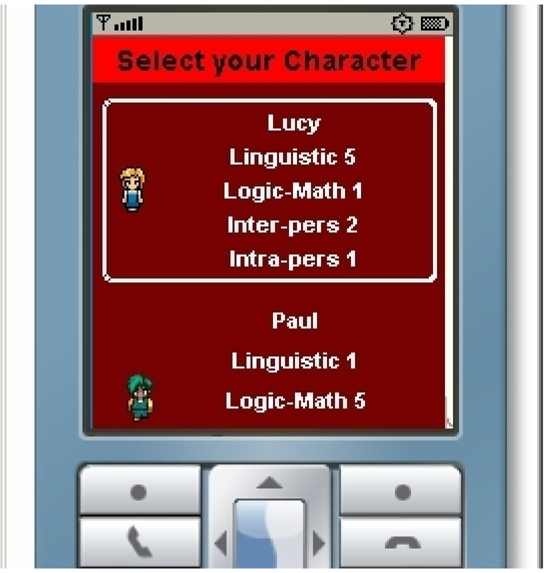
- Quiz: a text and one or more options to choose from.
- Decision Tree: similar to Quiz, however the different possible choices have no immediate reward but instead they link to a subsequent event (which can be any event type). This

allows construction of complex simulations of chains of choices or decisions.

- Conditional decision tree: is similar to Decision Tree, but some of the possible choices are available and visible to player only under particular conditions, typically a set of minimum points in one or more characteristics of the avatar (see later regarding characteristics).
- Simple: a text message that can be used as a "leaf" of a decision tree or as a simple random event.
- Multimedia: opens a multimedia resource and then links to a subsequent event. This can be used to enhance graphic aspect of the events or to insert audio/visual elements in the decision trees. Some basic file formats for images, audio and movie files are supported at the moment. It must be noted that current hardware limitations for mobile devices actually limit severely the quantity and size of the multimedia files that can be inserted in a whole game to a handful of low resolution images, but the game is ready to manage heavier duties in terms of multimedia content when mobile hardware will improve.
- Set internal variables value: the game holds an internal array. This can be used as a sort of "state machine" for complex events correlations. Up to 200 integer variables can be modified at run-time by use of this event. Putting this event at the end of a "decision tree" allows to keep memory of decisions taken by player.
- "Case of" tree: event structured just like the CASE instruction in programming languages. According to current value of one of the internal variables, different events can follow. This, in combination with the previously described event type, allows to set further events depending on previous decisions (maybe also decisions taken far earlier in game).
- Null event: game contents logic may require an "empty" event.
- Game Over: event overriding the normal game over rules.

Another important concept in the game, taken from the role-playing-games, is that the player's avatar is personalized by a set of 4 to 6 attributes, or "characteristics" (Kun, Byung 2007). The characteristics names are fully configurable: they could be, e.g., "Strength, Intelligence, Charisma, etc", if one wants to create a "typical" RPG avatar configurability. They may be "Linguistic, Logical-mathematical, Intrapersonal, Interpersonal" to follow Gardner's theory of Multiple Intelligences (Gardner 1983) in case of a career guidance game, where we're not interested in characterizing the physical aspects of the avatar but rather the facets of its intelligence.

Players can choose between different "characters" with varied skills (figure 1). During the game events, or better, choices made in "decision" events, can be set to modify these values in terms of rewards or penalties, so player's choices will have impact on characteristics' values and also possibly on game evolution, since with use of the above described "Conditional decision tree" events some options in game will be available only under certain conditions in player's characteristics.


Select your Character

Lucy
Linguistic 5
Logic-Math 1
Inter-pers 2
Intra-pers 1

Paul
Linguistic 1
Logic-Math 5

Figure 1: Game Screenshot. Choice of Player's Avatar at Beginning of a New Game.

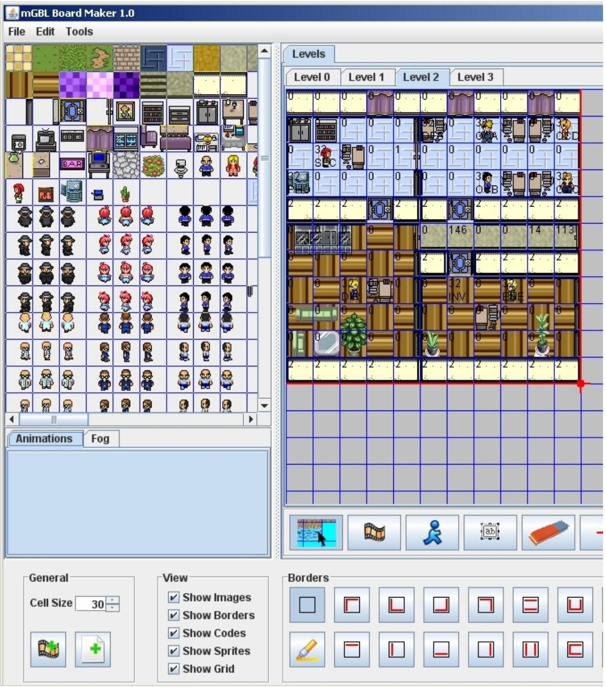
The Java code runs as a "game engine" while everything else is contained in "resource files". The process of compilation of the application with Java creates a "jar" archive file and a "jad" descriptor file. The jar file is the final "application" that will be installed on the mobile phone, it contains all the compiled code and all the resource files included in the Java project and its contents can be opened on any PC by using on of the many archives/compression management applications capable of opening such files. In particular, the resource files contained in the jar file include all the graphic resources used for game map creation or in multimedia events and three text configuration files containing, respectively:

- game setup: defining characteristics' names, game over conditions, etc;
- graphic setup: defining how one or more game maps (one per game "stage") will be constructed from the resources containing the sprites and graphic elements needed;
- events' contents: the complete list of all the events that may happen in game.

Game setup can be edited with a simple text editor.

For graphic setup, we developed for the task a Java tool called "Boardmaker" (figure 2) meant to be run on

PC, presenting a graphic user interface that allows relatively easily to create game maps and the corresponding "Graphic setup" text file. Different "levels" of the game can be created, each with its own map. Obstacles and borders can be inserted on the map: visible or invisible, permanent or removable, one way or two-ways. Also this tools allows to put on the maps the active "sprites" that will be the core of the game. The sprites will be lined to launching events, and in this case they will be assigned the (non-unique) three digit code that will "mark" the event or events associated to avatar's collision with the sprite. Sprite may also be used to give "keys" that allow removal of some obstacles or to give or take points to player. The tool also allows creation of mobile sprites by defining zero or more finite-states machines that will be included into the configuration file. Sprites associated with such tables can be programmed for movement on the map on fixed or random trajectories. Future developments in the code could allow sprites programmed to approach the player's avatar ("enemies to avoid") or to avoid it ("preys to catch"), but this is not in the list of priorities as this could be an interesting option for creation of more "arcade-style" games but is not particularly necessary for "learning" games.



Cell Size 30

☑ Show Images
☑ Show Borders
☑ Show Codes
☑ Show Sprites
☑ Show Grid

Figure 2: Boardmaker Tool Screenshot. Portion of the User Interface

For the list of events, XML is used (Natchetoi, Wu, Babin, Dagtas 2007). An ad hoc XML-DTD describes all the possible events' structures, so that events following this Data Type Definition will be forcibly consistent with game code. An XML-XSL file allows automatic conversion into the text file that must be inserted within the game resources. In this way a game creator can develop his own events for customized games with no need to know game code, as long as he follows the structure defined, and the use of XML allows automatic check of correctness of the structure. Yet it was not opportune to use directly the XML file in the resources, as the XML tags create a substantial overhead. Currently on most mobile phones the size limit of the jar file for installing the application is about 500 KB. To give more space for use of multimedia resources or for longer events lists, we opted for conversion of the list of the events into a "pure text" file, with events and their fields' contents separated by vertical bar character.

With all the above described elements, a wide range of game styles can be covered by the Mogabal engine. Some examples include:

- Quiz game: using "permanent" event-sprites, linked each to a long list of random quizzes regarding various topics.
- Exploration: use of fog of war and visible or hidden obstacles/borders can allow creation of labyrinth games of exploration.
- Arcade style: using mostly event-sprites with programmed or semi-random movement on a map, the game can be aimed to avoiding "baneful event" sprites while searching collision with "positive event" sprites.
- Simulation: an interactive map can "put" players avatar in a particular situation (e.g.: car accident or generally a crisis situation). Interaction with elements present may force player to try to make the right decision.
- Adventure: with more preliminary work of "plot creation", the simple "simulation" game above described can be evolved into a complex "adventure-game", with several stages (maps). By using the "Set internal variables value" event it is possible for the game to "keep memory" of players choices and have the "adventure world" react accordingly.

In all these different "game styles" contents are completely customizable. As an example: in a quiz game, as above described, the lists of quizzes could be substituted with others regarding completely different topics by simply changing the "events' content" configuration file while maintaining unchanged game logic and graphic appearance.

5. CASE STUDY: "MOGABAL E-HEALTH" GAME IN MGBL 2ND USER TRIAL

In mGBL project 3 cycles of User Trials were planned, directly related to the iterative process adopted for mGBL design and development.

The second User trials took place in September 2007, finishing at end of year 2 of the project. They were conducted with "Focus Groups" methodology and primarily sought understanding and insights from members of target audiences concerning the 3 mGBL games in terms of "Fun and Playability" and "Content". Opinions from potential users (both students and teachers) were sought regarding style and design, relevance to topic, suitability, support to users in developing decision-making skills.

Six different User Trials were conducted in four countries, evaluating the three game models developed within the project. The Mogabal game engine was tested in four of such trials and with different game templates, regarding the topics of e-health, e-guidance and maritime distress. Thus the above described instruments of "customization" were used to create three different games quite different in game style and completely different in educational contents.

Focus of this paper is the game prototype we developed for e-health contents, as it is a good example of possible use of such mobile learning games for use in simulation of crisis situations.

The game was developed, for what regards the "game-style", as a combination of Simulation, Quiz and Adventure. The "characteristics" of the avatar were of less importance in this context and this feature of the game was not used.

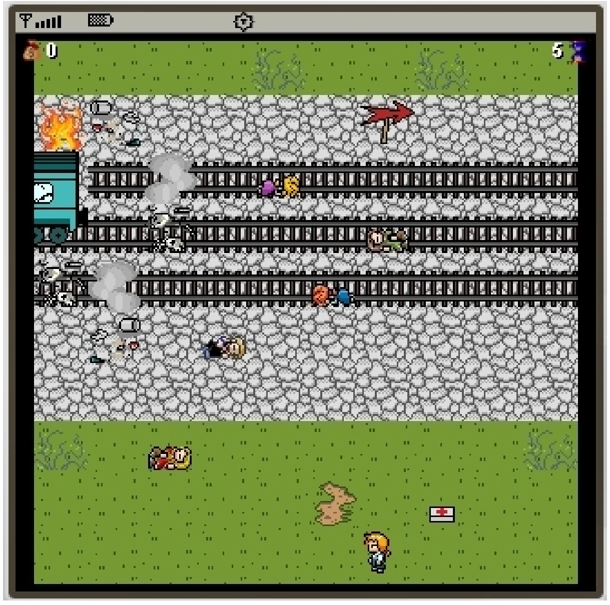


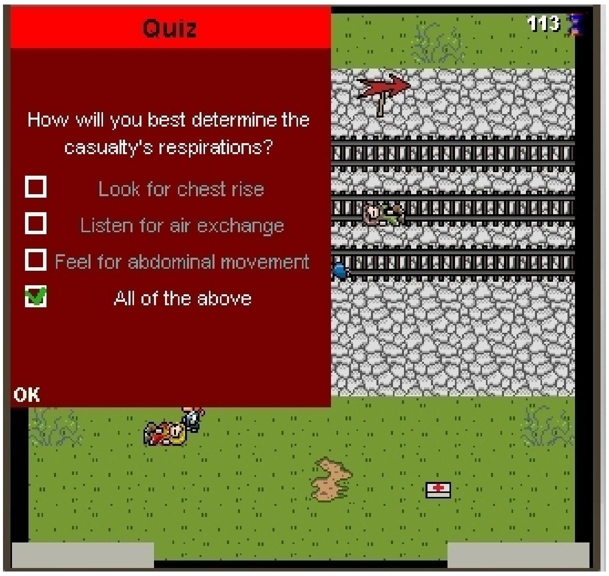
Figure 3: Game Screenshot. Triage on Train Crash Site.

In the first stage of game (figure 3), the player has to conduct a triage on the victims of a train accident, that is, he has to approach the injured and follow procedures that allow him to discriminate and "tag" them with colour codes that will help oncoming medical services to treat victims in order of urgency (figure 4). Only under particular circumstances in the triage also some medical assistance is given: objective is only quick decision on priority of treatments.

The second stage of the game revolves instead on use of first aid techniques as the player has to assist the victims of a road accident (figure 5).

Touching the casualties, the system displays events and series of linked multiple answer questions forming decision trees.

The game contents for first stage of game (Triage) are quite specialized and the prototype game was created as an example for a possible tool for knowledge assessment in combination with a triage course for emergency operators, to help in memorization of correct procedures.



Quiz

How will you best determine the casualty's respirations?

☐ Look for chest rise
☐ Listen for air exchange
☐ Feel for abdominal movement
☑ All of the above

OK

Figure 4: Game Screenshot. First Decisions in Triage.

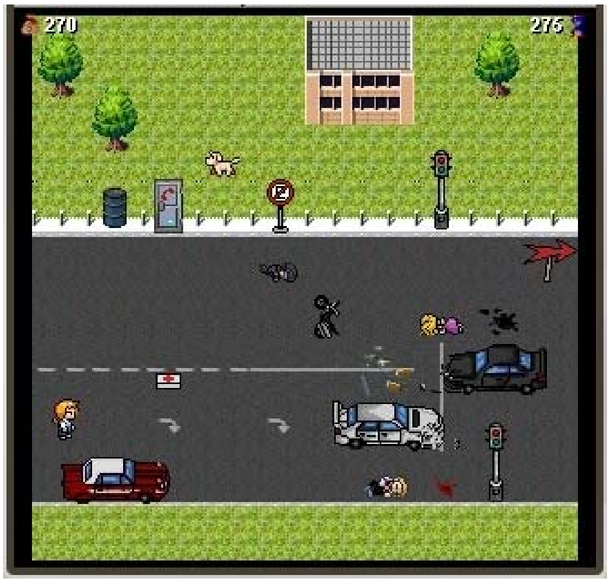
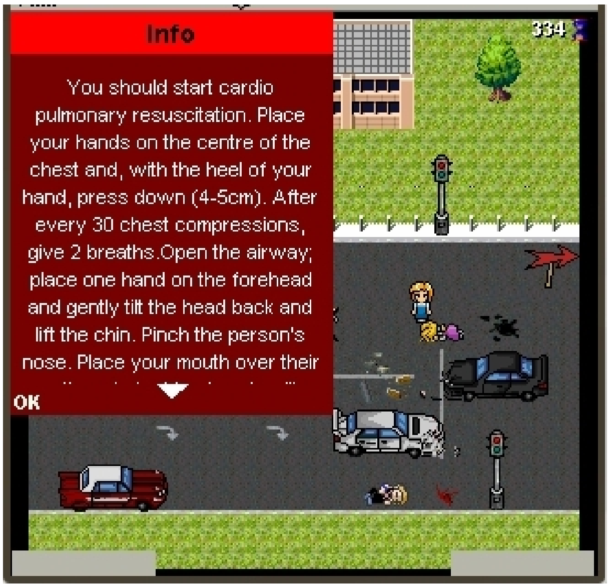


Figure 5: Game Screenshot. Second Level. First Aid After a Car Crash.

The second stage of the game can be an example of educational game: playing the game several times and receiving feedback (figure 6), after taking a decision, indicating what was the correct thing to do, means that a young player who will reach the "highest score" will probably have also fixed in memory the correct first aid procedures, paying to them more attention than if he had to memorize them in more "standard" learning ways.

Reception from audience in the focus groups (which were composed by teachers and students) was quite positive. There was overall appreciation for game playability and fun by part of the students and for its potential developments and educational possibilities by part of the teachers.



You should start cardio pulmonary resuscitation. Place your hands on the centre of the chest and, with the heel of your hand, press down (4-5cm). After every 30 chest compressions, give 2 breaths.Open the airway; place one hand on the forehead and gently tilt the head back and lift the chin. Pinch the person's nose. Place your mouth over their

Figure 6: Game Screenshot. Feedback and Precious Information for the Player.

6. FUTURE POSSIBILITIES AND DEVELOPMENTS

Although the “Mogabal e-health” game developed for and tested during the mGBL 2^{nd} user trials was a sort of limited prototype, acceptance received show a good potential of this kind of mobile game for learning in the field of crisis simulation.

In particular, the game engine structure, as shown in chapter 4, currently still suffers from the hardware limitations. As mobile phones will allow more and more to use screens with better resolution and a larger memory footprint that will allow to insert into the games larger maps to be explored and more multimedia elements like photos, videos and so on, so will become better the chance to create “interesting” games-simulations that can help in the training and in developing decision-making skills.

Work will continue in creating new prototypes and all the Mogabal source code and related material (documentation, the Boardmaker tool) will be published as Open Source under European Union Public Licence (EUPL) v1.0. Hopefully this will lead to the birth of a larger community of developers, both for improvement of the code with further features and improved bug-tracking (thanks to a larger audience) and for creation of more other game examples exploring the full range of possibilities given by Mogabal game engine.

REFERENCES

Argyris, C., 1976. Increasing Leadership Effectiveness, Wiley, New York.

Bellotti, F., Berta, R., De Gloria, A., Ferretti, E., Margarone, M., 2003. Designing mobile games for a challenging experience of the urban heritage. *Proceedings of EURO-PAR 2003 Parallel Processing, Lecture Notes in Computer Science,* Vol 2790 pp. 1129-1136. August 26-29, Klagenfurt (Austria).

European Union Public Licence official web site at: http://ec.europa.eu/idabc/eupl.

Fabricatore, A., 2000. Learning and Videogames: an unexploited synergy. Available from: http://www.learndev.org/dl/FabricatoreAECT2000.PDF Accessed 27th April 2008.

Gardner, H., 1983. Frames of Mind. Basic Books, New York.

Ha, I., Yoon, Y., Choi, M., 2007. Determinants of adoption of mobile games under mobile broadband wireless access environment. *Information & management* 44 (3): 276-286.

Kolb, D.A., 1984. Experiential Learning, Prentice-Hall, New Jersey.

Kun, C. L., Byung, S. M., 2007. Enhanced Avatar Design Using Cognitive Map-Based Simulation. CyberPsychology & Behavior 10 (6): 757-766.

Lei, Y., Hui, L., 2006. Which One Should be Chosen for the Mobile Geographic Information Service Now, WAP vs. i-mode vs. J2ME. Mobile networks and applications : the journal of special issues on mobility of systems, users, data and computing 11 (6): 901-916.

mGBL official web site at: http://www.mg-bl.com.

Natchetoi, Y., Wu, H., Babin, G., Dagtas, S., 2007. EXEM : Efficient XML data exchange management for mobile applications. Information Systems Frontiers 9 (4): 439-448.

Race, P., 1994. *The Open Learning Handbook.* London: Kogan Page.

Vygotsky, L.S., 1978. Mind in society: The development of higher psychological processes (Cambridge, MA: Harvard University Press.).

AUTHORS BIOGRAPHY

Stefano Mininel holds a PhD degree in Bioengineering and is a research assistant at the Department DEEI of the University of Trieste, Italy, since 2006.

Federica Vatta holds a PhD degree in Bioengineering and is an assistant professor of Bioengineering at the Department DEEI of the University of Trieste, Italy, since 2005.

Sara Gaion holds a M.Sc. Degree in Biomedical Engineering and is currently a PhD student in Information Engineering at the Department DEEI of the University of Trieste, Italy.

Walter Ukovich is Full Professor of Operations Research at the Faculty of Engineering of the University of Trieste. He is the Director of the Department DEEI and the Coordinator of the PhD School in Information Engineering of the University of Trieste. His scientific interests are in management of distribution networks, organization and management of health services, programming and production control, logistics, innovation and evaluation.

A HIGH PERFORMANCE COMPUTING-BASED APPROACH FOR THE REALISTIC MODELING AND SIMULATION OF EEG ACTIVITY

Vatta F. [(a)], **Mininel S.** [(a)], **Bruno P.** [(a)], **Meneghini F.** [(a)], **Di Salle F.** [(b,c)]

[(a)]DEEI, University of Trieste, Trieste, Italy
[(b)] Department of Neurosciences, University of Pisa, Pisa, Italy
[(c)] Department of Cognitive Neurosciences, University of Maastricht, Maastricht, The Netherlands

[(a)]{vattafe, mininel, bruno, meneghini}@deei.units.it [(b, c)]francesco.disalle@psychology.unimaas.nl

ABSTRACT

An original simulation framework specifically conceived and designed to achieve high performance three-dimensional (3D) realistic modeling and simulation of electro-encephalographic (EEG) brain activity, named TEBAM (True Electrical Brain Activity Mapping), is presented. We describe the integrated ICT framework that has been proposed and developed for TEBAM, specifying the design characteristics, implementation and tools interconnections. TEBAM relays on patient's specific realistic head modeling, based on identification of the various head structures necessary for an accurate model building by means of suitable clinical imaging protocols presented in this paper. TEBAM is implemented and optimized with a very flexible approach to solve in short time, by means of High Performance Computing resources, the large scale computations needed. 3D simulation results can be visualized in TEBAM framework in different multimodal ways, combining the anatomical information with the computed results to give an optimal insight of computation output, relying also on stereographic visualization.

Keywords: High Performance Computing, EEG simulation, realistic head modeling, multimodal neuroimaging

1. INTRODUCTION

Mathematical models of generation of electro-encephalographic (EEG) brain potentials are a powerful and helpful tool to better analyze and understand the mechanisms involved in the development of brain activity in normal or pathological conditions. By means of modeling and simulation in computers, the neural sources of the scalp recorded EEG potentials can be non-invasively estimated and imaged, and the origin and evolution of brain activity can be first studied *in silico*, where hypotheses can be formulated and studied prior to their validation in vivo, thus reducing the requirements of many complex intrusive techniques.

A key point towards the simulation and visualization of neural sources of EEG brain activity within the specific patient's head with both high spatial and temporal resolution is the multimodal integration of EEG and clinical imaging data, as the former allow measurement of EEG activity with an optimal temporal resolution while the latter are characterized by a very high spatial resolution (Baillet, Mosher, and Leahy 2001). The EEG inverse problem is the process of estimating the characteristics of the neural sources responsible for a given EEG distribution measured at the scalp electrodes. This can be achieved by means of iterative computational methods with a large number (several hundreds) of iterative EEG forward simulations to find the optimal source parameters corresponding with the measured EEG potentials (Baillet, Mosher, and Leahy 2001). To accomplish this non trivial task, a suitable simulation framework should be available. First of all, a precise and realistic representation of the electrical properties of the specific subject's head, in terms of shape and electric conductivities, is necessary to achieve an accurate EEG forward simulation (Baillet, Mosher, and Leahy 2001). Moreover, the adopted head model should also be able to incorporate various sets of tissues with different conductivities (Vatta, Bruno, and Inchingolo 2005). This is extremely important in clinical applications in which also pathological formations as brain lesions (which are characterized by a large variability in shape and conductivity) have to be included in the head model (Vatta, Bruno, and Inchingolo 2002). Once built, realistic head models require the use of demanding numerical computer methods for EEG forward problem solution and hence for electrical brain activity mapping (Baillet, Mosher, and Leahy 2001). A suitable, flexible and performing simulation framework should therefore account for all these constraints.

In this paper, an original simulation framework named TEBAM (True Electrical Brain Activity Mapping) is presented. TEBAM was specifically designed and implemented to account for all the above mentioned constraints. In the following sub-sections are presented the design specification, the imaging protocols requirement for accurate head modeling, the structure and implementation of TEBAM followed by the validation and testing of the framework.

2. SIMULATION OF THE EEG POTENTIALS GENERATION

The EEG forward problem, which has to be iteratively solved in TEBAM's framework for electrical brain activity mapping, is governed by Poisson's differential equation (Bronzino 1985)

$$\nabla \cdot \left(\sigma \nabla \Phi\right) = \nabla \cdot \vec{J}_i = \rho \qquad (1)$$

where $\vec{J}$ i is the applied current density of the neural brain source (A/m^2), σ is tissue electrical conductivity (Ωm)$^{-1}$, and Φ is the electric potential in the problem domain. Realistic head models impose numerical computational methods for the solution of eq. 1, as the Finite Difference Method (FDM), which has been implemented in TEBAM framework thanks to its characteristics of flexibility which also allow an easy implementation of anisotropic electrical conductive domains. This typically involves the solution of a large and sparse linear algebraic equations system (Ax=b). Hence, the main characteristics of the bioelectrical problems computations in TEBAM framework are: 1) Large-scale, i.e., large memory and CPU time requirements; 2) Iterative, as electrical brain activity mapping requires EEG forward problem solution to be performed iteratively; 3) Multistep, as simulations are typically composed of a fairly complex steps sequence that are arranged in pipeline and classified as modeling, simulation computing and visualization.

The TEBAM pipeline is composed by 5 steps: 1) Construction of a model of the physical problem domain, in terms of shape and physical properties, given by the patient-specific volume of the head (Baillet, Mosher, and Leahy 2001); 2) Application of boundary conditions and/or initial conditions, as source modeling and specification of initial data for the iterative computations are required; 3) Computing, as EEG forward and inverse solutions can be computed by solving a linear system of algebraic equations, derived from the numerical solution of eq. 1; 4) Validation and test of the results, as during the development phase results correctness has to be checked upon simple physical test domains for which independent solutions methods are available; 5) Visualization, as simulation results have to be visualized by means of suitable Scientific Visualization tools (Schroeder, Martin, and Lorensen 1996). Fig. 1 shows an example of result of the computational process.

TEBAM was designed as an integrated framework in which visualization is linked with computation and geometric design to interactively explore (steer) a simulation in time and/or space. In synthesis, the TEBAM problem solving framework has been designed to address the following issues: 1) Integration in data collection of multimodal anatomo-functional data; 2) Integration in data analysis, as modeling, simulation and visualization aspects of the problem have to be used in chorus; 3) Interactivity, to understand cause-effect relationships; 4) Extensibility, to get not a monolithic solution for one problem but possibility of reuse for solving also new problems; 5) Scalability, as although a full EEG inverse problem solution in short time requires the use of High Performance resources, tools can be run even on high-end PCs.

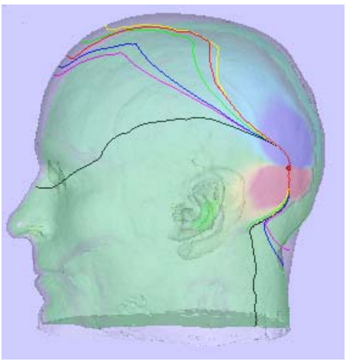


Figure 1: Visualization of current lines originated by a neural EEG dipolar source located approximately in visual cortex.

3. ARCHITECTURE OF THE SYSTEM

TEBAM provides an optimized dataflow programming framework, based on modules which implement components for computational, modeling and visualization tasks to build an interactive framework in which the researcher is free to change various parameters as mesh discretization, iterative solution method, neural source placements and visualization tools displayed.

The main bricks of TEBAM are: 1) building of the patient-specific realistic head model; 2) numerical EEG forward and inverse problem solution, with multiple iterative forward solutions; 3) visualization of the computed results.

As first step, a 3D voxel matrix is created, modeling the volume conductor of the head of the specific patient under analysis. This is done with segmentation of a suitable set of clinical images of the subject's head by means of 3D Slicer (3D Slicer UG 2006) and then assigning a scalar or a tensorial conductivity value to each identified pixel, according to the isotropic or anisotropic conductivity of the specific head model compartment (Bruno et al. 2006). The needed requirements in terms of clinical imaging protocol for accurate head modeling are discussed in Section 4.

The second step implies the building and solution of the large and sparse linear algebraic equations system (Ax=b) derived from the numerical FDM discretization of eq. 1. TEBAM framework has been designed to build and solve efficiently the equations system of step 2, giving high flexibility in the choice of solution methods and being able to run with small modifying either on mono-processor PC or, in parallel, upon large High Performance Computing (HPC) Systems. HPC resources are an adequate instrument for a consistent reduction in solution time for solving of large scale problems, as the computational load is subdivided using more CPUs and inter-CPUs communication is managed by MPI (Message Passing Interface). The need for code parallelization and for the use of HPC in TEBAM was due to the magnitude of the problems addressed. In fact, a conductive head model derived from segmentation of a series of MRI images with adequate spatial resolution leads to a linear equation system with millions of unknowns for the solution of a single EEG forward problem. As the EEG inverse problem solution requires several iterative EEG forward problems solutions, HPC becomes then mandatory to reduce computation times especially for clinical applications purposes. In TEBAM a typical parallelization strategy, named “divide and conquer”, has been adopted. Each CPU solves the problem in its sub-domain and MPI is used to exchange values necessary to each CPU for contour values (see Fig. 2). A specifically designed application was written in C++, compiled in Visual C 6.0 and in gcc 3.0 frameworks to build up a multi platform application, capable of running on either windows or linux machines. Libraries rely upon wxWidgets (Smart, Hock, and Csomor 2005), freeware and open source multiplatform library to help in creation of graphic user interfaces (GUI) and in several other tasks, VTK (Schroeder, Martin, and Lorensen 1996) for head model data reading and for all the interactive 3D visualization pipeline and Petsc (Balay et al. 2002) for linear system solution and parallelization issues. The solution application uses the PETSc libraries for twofold reasons: to create an open-source tool entirely based upon open-source libraries and because these libraries allow a high level of abstraction to leave “transparent” the low level calls and message exchanges between CPUs, hence allowing focusing on optimization and search for stable and accurate solution methods. The third step, visualization, is described in Section 6.

4. IMAGING PROTOCOL FOR ACCURATE HEAD MODELING

Accuracy achievable in EEG source imaging is influenced by errors committed in head modeling (Vatta, Bruno, and Inchingolo 2002). Clinical images, typically MRI and CT, are used for head model building. Head modeling accuracy mainly relies on correct identification, by image segmentation, of head structures characterized by different electrical conductivities to be modeled as separate compartments, assigning each an appropriate conductivity value (Vatta, Bruno, and Inchingolo 2002). Brain lesions show large variability and an intrinsic difficulty for segmentation (Vatta, Bruno, and Inchingolo 2001); hence, acquisition of finely tuned images (e.g., MRI with contrast medium injection) is often required, but this kind of images is not the best also for identification of standard head structures as scalp, skull, etc. The possibility of deriving information about tissue anisotropy from clinical images is also desirable (Vatta, Bruno, Di Salle et al. 2008). Notably, the MR-based diffusion tensor imaging (DT-MRI) has recently been suggested to map the conductivity tensor of the brain given the high correlation between electrical conductivity tensor and water self-diffusion tensor, with the potential to further refine the head modeling by taking the anisotropy of white matter into account. In general, one imaging procedure giving best results in some conditions, e.g., for identification by its image contrast of a specific head structure, may not be the optimum in other situations.

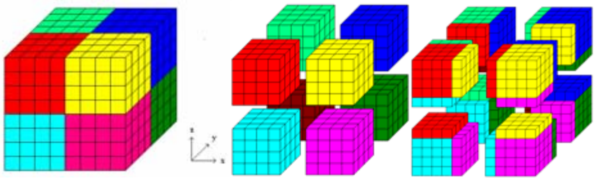


Figure 2: Parallelization: the problem domain is divided in a suitable number of sub-domains (left). Each CPU is assigned a sub-domain to be solved locally (center). A number of “ghost points” must be created in each sub-domain, where values computed by other CPU will be stored (exchanged by MPI) to give correct border values to each sub-domain (right).

The available clinical imaging protocols used for the purpose of clinical morphological analysis have been analyzed from a segmentation point of view, to define the procedures most suitable for accurate identification, also in the presence of pathology, of the head structures necessary for head modeling, also accounting for the above described modeling issues. The following sets of clinical images have been analyzed: Proton Density, FLAIR T2, Inversion Recovery, Spin Echo with contrast medium injection, Spin Echo DP/T2, Spin Echo T1, T2 dry, Turbo SE T2 and CT. The following head model compartments have been identified by means of image segmentation: skin, fat tissue, skull, cerebrospinal fluid (CSF), ventricles, gray matter (GM), white matter (WM), medulla and cerebellum, eyes, muscle, internal air and brain lesions. Results of segmentation applied to the adopted image sets to identify the above listed head structures for head modeling purposes demonstrate that an appropriate multi-modal image set has to be acquired for accurate model compartments identification. Tab. 1 summarizes, for each image set, tissues identifiable by segmentation with a qualitative evaluation referenced to an anatomical brain atlas. A protocol has then been identified and proposed for acquisition of multi-modal patient’s specific imaging data, to be integrated for head model building for EEG brain activity mapping (Vatta, Bruno, Di Salle et al. 2008). The best imaging sequences, among the ones adopted in clinical environment, for the identification of the different head tissues which have to be included in the head model, are summarized in Tab. 2.

The performed studies allowed the identification of a multimodal clinical imaging protocol for the acquisition of patient’s data to be integrated for building an accurate volume conductor head model. Contrarily to imaging protocols for sole diagnostic clinical purposes, image acquisition should be performed with a spatial resolution constant in the 3 scan dimensions or at least similar, to attenuate the loss of information due to

Table 1: Quality of segmentation for head tissues in different image sets

Image set	Identifiable tissues	Segmentation quality
Proton Density	GM, WM, ventricles, CSF, eyes	medium
FLAIR T2	CSF, ventricles, skin	excellent
Inversion Recovery	GM, WM, ventricles, CSF, skin	excellent
Spin Echo + contrast medium	ventricles, CSF, eyes, skin	depending on acquisition
Spin Echo PD/T2	CSF, ventricles, eyes	good
Spin Echo T1	GM, WM, ventricles, CSF, eyes, fat	medium
CT	Skull	excellent
T2 Dry	CSF, eyes, glioblastoma	medium
Turbo Spin Echo T2	CSF, eyes, abscess	medium

Table 2: Head tissues and optimal image sets for their identification

Tissue	Image set
GM	Inversion Recovery
WM	Inversion Recovery
CSF	Inversion Recovery
Medulla and cerebellum	Inversion Recovery
Soft bone	CT
Hard bone	CT
Muscle	FLAIR T2
Fat	Spin-Echo T1
Skin	Inversion Recovery, FLAIR
Soft tissue	Proton Density, Spin Echo T1
Internal air	Cannot be separated from bone in MR, cannot be visualized in CT
Eye	T2 dry, Turbo Spin Echo T2

pixels' interpolation between adjacent sections in the 3-D model building-up. CT acquisition should be performed as follows: 1) Acquisition of contiguous slices of reduced thickness (5mm, 3mm better); 2) Acquisition volume, preferably unique and uniform, extending downwards from head vertex to include at least the skull base (first cervix vertebra desirable); 3) Matrixes 512*512. For MRI acquisitions: 1) Contiguous slices of reduced thickness (2 mm or less); 2) Matrixes 512*512. Larger image matrixes and reduced gap between adjacent slices allow higher spatial resolution in models obtained. For accurate head model geometrical definition, the proposed image acquisition protocol provides using the following multimodal scans: CT for skull; MR Inversion Recovery for GM, WM, cerebellum and CSF; FLAIR T2 for muscles and skin; Spin-echo T1 for fat tissue, para-nasal sinuses and brain lesions; Turbo-spin-echo T2 for eyes. Suitable DT-MRI sequences have to be used for information about tissue anisotropy. Distance between adjacent slices should be better limited to 2-3 mm, possibly covering a volume extended from head vertex to the first cervical vertebra. DT-MRI acquisition can be limited to the reduced volume containing the anisotropic tissue under analysis. Resolution requirements are determined by the most demanding modality (DT-MRI), while field of view (model extension) by model completeness.

5. RESULTS

The 3D EEG simulation framework of TEBAM has been validated by means of EEG forward problem solution using a spherical head model for which analytical solutions were available (Vatta, Bruno, and Inchingolo 2005), using the successive over-relaxation (SOR) method. Optimization analysis has been performed to improve code performance regarding both sequential solution and parallelization procedures. PETSc libraries give excellent profiling instruments that allow evaluation of the optimization degree reached by the use of several CPUs in parallel framework. Fig. 3 shows an example of optimization results related to an EEG forward problem solution with a conductive head model matrix of 64x64x115 elements, a low resolution model used for the sake of testing purposes. Simulations were performed with IBM SP5 made kindly available by the Interuniversity Consortium CINECA (Bologna, Italy) to test the performance of the HPC applications presented in this paper.

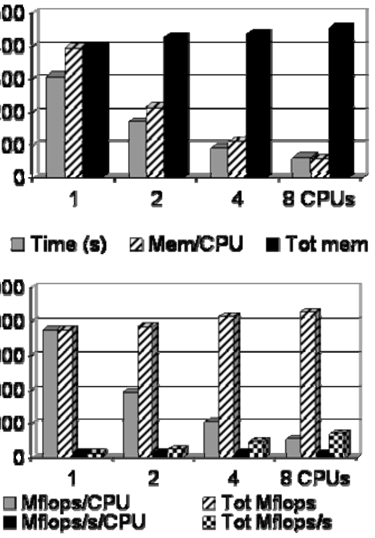
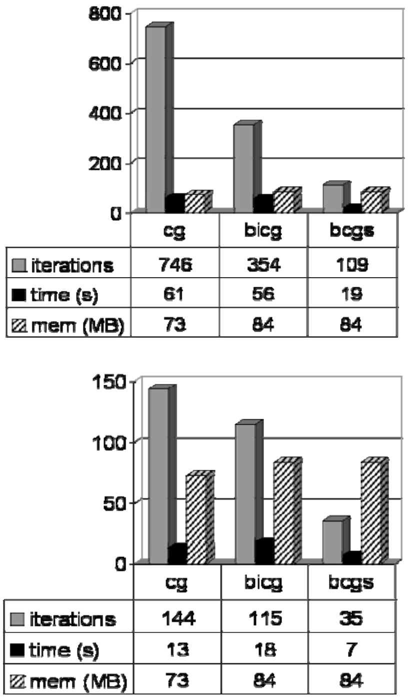


Figure 3: Compared performances with 1, 2, 4 or 8 CPUs on CINECA IBM SP5. (Top) solution times (in seconds) and memory used (in MB); (Bottom) number of floating point operations (in Mflops) by each CPU and for whole problem solution.

PETSc libraries give also a good flexibility and easiness in the choice of suitable iterative solution methods and error tolerances. Next optimization step was then the search for solution methods and tolerances able to guarantee the best performances without sacrificing accuracy in EEG forward problem solution and in 3D EEG source reconstruction. Tests were carried upon a conductive head model constructed out from segmentation of a set of 115 MRI sagittal 256x256 scans. The obtained 3D conductivity matrix (the head model) was sub-sampled to two volumes with lesser resolution to reduce computational load during tests. The following iterative solution methods have been tested and analyzed: Successive Over-relaxation (SOR); Symmetric SOR (SSOR); Conjugated Gradients (CG); Bi-Conjugated Gradients (BiCG); Squared Bi-Conjugated Gradients (BCGS). Different tolerance criteria were examined as parameter for choice of stopping iterative solution, with tolerance values for relative error norm ranging from 10^{-6} to 10^{-12}. Comparisons between three iterative methods are shown in Fig. 4, for an EEG forward problem simulation on a 64x64x115 head model on a mono-processor system (AMD Athlon XP, 2,2 GHz). Tables show performance comparison of the three methods in terms of iterations number needed to reach the required tolerance, solution time and memory needed. In this problem the CG method converges in a larger iterations number but with less memory needs and in shorter time than BiCG. BCGS show the best performances in solution time but with larger memory requirement. The optimization and parallelization procedures lead to a large improvement in the performance, shortening computational time from 45 to less than 1 minute (forward problem solution on a 128x128x115 model).



	cg	bicg	bcgs
▨ iterations	746	354	109
■ time (s)	61	56	19
▨ mem (MB)	73	84	84

	cg	bicg	bcgs
▨ iterations	144	115	35
■ time (s)	13	18	7
▨ mem (MB)	73	84	84

Figure 4: Performance comparison with different iterative solution methods (cg = conjugate gradients; bicg = bi-conjugate gradients; bcgs = squared bi-conjugate gradients). (Top) results for reaching a tolerance of 10^{-7}; (Bottom) tolerance of 10^{-6}.

6. VISUALIZATION

The visualization pipelines developed for TEBAM make full use of several data-fusion techniques (see Fig. 5, in which are shown some examples of EEG forward solutions computed on high resolution head models) and of 3D stereographic rendering and have been

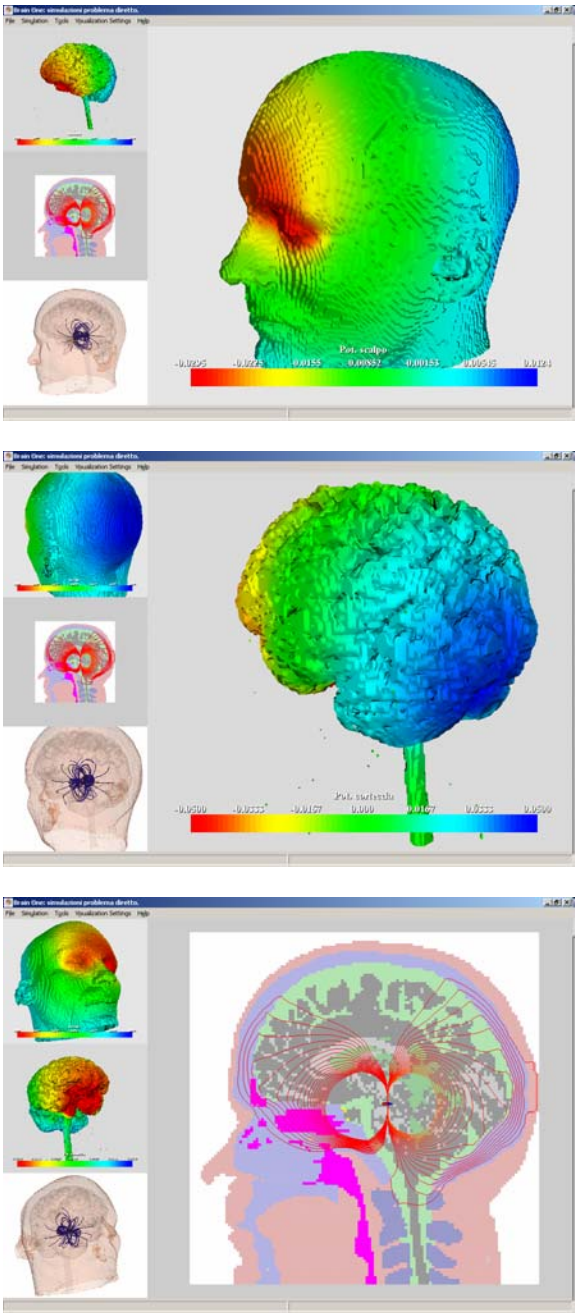


Figure 5: (Top) Scalp surface with electric potential color-map; (Middle) Cortex surface with electric potential color-map.; (Bottom) Tissues cut plane with potential iso-lines.

developed using VTK libraries (Schroeder, Martin, and Lorensen 1996). The hardware stereo support used for testing is an auto-stereo display DTI 2015XLS Virtual Window (Dimension Technologies Inc.) based on Parallax Illumination technology.

The visualization module of TEBAM focused on visualization techniques useful to help data analysis in the context of anatomo-functional integration. The objective in developing these visualization instruments was to have a tool for a better "intuitive" understanding of the 3D EEG source reconstruction procedures, both for research purpose and for future users or developers of TEBAM tools. Visualization output is divided in 4 panels (see Fig. 5), each with a different rendering showing different features.

This multimodal data presentation helps understanding the link between functional and anatomical data. In all the four graphics visualizations, the user can freely "navigate" the model using the mouse to rotate, zoom and pan. The main rendering panel may be switched to stereo 3D mode to improve comprehension of complex configurations adding the depth clues. Most visualization parameters may be changed at will by the user, to allow a deep and meaningful "neuro-navigation".

7. CONCLUSION

The TEBAM original simulation framework presented in this paper is a powerful tool to model and simulate brain activity with high spatio-temporal resolution and accuracy.

TEBAM's features allow overcoming many important limits of several scientific and commercial software. Qualifying features are: flexibility in computational methods, flexibility in modeling to accurately conforming to the specific patient's head, scalability from PC to HPC, multimodal stereo visualization.

ACKNOWLEDGMENTS

Work supported by University of Trieste–Young Researchers Project - Università degli Studi di Trieste-Progetto Giovani Ricercatori, Trieste, Italy and by the Interuniversity Consortium CINECA, Casalecchio di Reno (BO), Italy.

REFERENCES

Baillet, S., Mosher, J.C., Leahy, R.M., 2001. Electromagnetic brain mapping. *IEEE Signal Processing Magazine*, 18(6), 14-30.

Balay, S., et al., 2002. *PETSc users manual, Technical Report* ANL-95/11 Revision 2.1.5, Argonne National Laboratory.

Bronzino, J.D., Ed., 1985. Numerical methods for bioelectric field problems. In: *Biomedical engineering handbook*, Boca Raton, FL: CRC, 161-188.

Bruno, P., Hyttinen, J., Inchingolo, P., Magrofuoco, A., Mininel, S., Vatta, F., 2006. A FDM anisotropic formulation for EEG simulation. *Proceedings 28th Annual International Conerence. IEEE-EMBS*, pp. 1121 – 1125, New York, USA.

Cuffin, B.N., 2001. Effects of modeling errors and EEG measurement montage on source localization accuracy. *Journal of Clinical Neurophysiology*, 18, 37-44.

Schroeder, W., Martin, K., Lorensen, B., 1996. *The Visualization Tool-kit: An Object-oriented Approach to 3D Graphics*. Prentice-Hall, NJ.

Smart, J., Hock, K., Csomor, S., 2005. *Cross-Platform GUI Programming with wxWidgets*, Prentice Hall, NJ.

3D Slicer Users Guide. Available form: http://www.slicer.org

Vatta, F., Bruno, P., Inchingolo, P., 2001. Influence of lesion geometry estimate on EEG source reconstruction. *IFMBE Proceedings*, pp. 974-977. vol. 1, Medicon 2001, Pula, Croatia.

Vatta, F., Bruno, P., Inchingolo, P., 2002. Improving lesion conductivity estimate by means of EEG source localization sensitivity to model parameter. *Journal of Clinical Neurophysiology*, 19, 1–15.

Vatta, F., Bruno, P., Inchingolo, P., 2005. Multiregion bicentric-spheres models of the head for the simulation of bioelectric phenomena. *IEEE Transactions on Biomedical Engineering*, 52, 384–389.

Vatta, F., Bruno, P., Di Salle, F., Esposito, F., Meneghini, F., Mininel, S., Rodaro, M., 2008. Head modeling for realistic electrical brain activity mapping: identification of a multimodal neuroimaging protocol. *Biomedical Sciences. Instrumentation*, in press.

AUTHORS BIOGRAPHY

Federica Vatta holds a PhD degree in Bioengineering and is an assistant professor of Bioengineering at the Department DEEI of the University of Trieste, Italy, since 2005.

Stefano Mininel holds a PhD degree in Bioengineering and is a research assistant at the University of Trieste, Italy, since 2006. His research interests are in scientific visualization and modeling of biological systems.

Paolo Bruno holds a Ph.D. degree in Bioengineering and is with the Italian Ministry of University and Research (MIUR) since 2003. His research interests are in modelling of physiological systems and bioelectric problems.

Fabio Meneghini is a PhD student in Bioengineering at the University of Trieste, Trieste, Italy, working on EEG modeling and on high performance computing.

Francesco Di Salle is Full Professor of Methods of Neuroimaging at the Faculty of Psychology of the Maastricht University (The Netherlands) and Associate Professor of Neuroradiology at the University of Pisa (Italy).

MODELING OF MARINE SYSTEMS AND PROCESSES AIMED AT OPTIMAL SHIP HANDLING

Enco Tireli[(a)] **, Josko Dvornik**[(b)] **, Srđan Dvornik**[(b)]

[(a)]University of Rijeka
Faculty of Maritime Studies
Studentska ulica 2
51 000 Rijeka
+385 51 338 411

[(b)]University of Split
Faculty of Maritime Studies
Zrinsko-frankopanska 38
21 000 Split
+385 21 380 762

[(a)] tirelli@pfri.hr, [(b)]josko@pfst.hr, [(b)]sdvornik@pfst.hr

ABSTRACT

The aim of this paper is to demonstrate the successful application of system dynamic simulation modelling at investigating performance dynamics of the marine steam turbine in load conditions, in the example of load of marine synchronous generator. Marine steam turbine at the load of synchronous generator is a complex non-linear system, which needs to be systematically investigated as a unit consisting of a number of subsystems and elements, which are linked by cause-effect (UPV) feedback loops (KPD), both within the propulsion system and with the relevant surrounding. Marine steam turbine will be presented by a set of non-linear differential equations, after which mental-verbal structural models and flowcharts in System dynamics symbols [1 and 2] will be produced, and the performance dynamics in load condition will be simulated in DYNAMO simulation language.

The results presented in the paper have been derived from the scientific research project „SHIPBOARD ENERGY SYSTEMS,
*ALTERNATIVE FUEL OILS AND REDUCTION OF POLLUTANTS EMISION" supported by the Ministry of Science, Education and Sports of the Republic of Croati*a.

Keywords: steam turbine, synchronous generator, simulation modelling, simulation

1. SIMULATION MODELLING OF MARINE STEAM TURBINE

1.1. Mathematical model of marine steam turbine

Figure 1. shows (Isakov and Kutljin 1984), a model of marine steam turbine machinery which drives electric synchronous generator.

In the presented case there are two essential situations of ability of energy accumulation:

1. in steam volume (steam area, steam volume of the turbine) and
2. in the turbine rotor,

while the main condenser is observed as a special governing object.

Each of the stated parts is described by its mode equation, that is, by the differential equation which describes the performance dynamics.

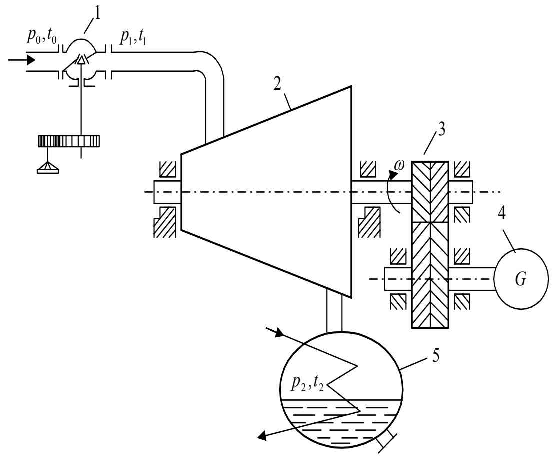


Figure 1: Steam Condensation Machinery of the Marine Turbine Generator
1- governing valve, 2- turbine, 3- reduction gear,
4- generator, 5- condenser

System dynamic mathematical model of the marine steam turbine is defined by means of explicit form of differential equation, or terms, (Isakov and Kutljin 1984):

Equation of the turbine steam volume

$$\frac{d\psi_1}{dt} = \frac{\mu}{R_\mu} + \frac{\Psi_0}{R_{\psi 0}} - \frac{\Psi_1}{R_{\psi 1}} \tag{1}$$

Equation of the turbine rotor dynamics

$$\frac{d\varphi}{dt} = \frac{\psi_1}{T_{\psi 1}} - \frac{\psi_2}{T_{\psi 2}} - \frac{\varphi}{T_\varphi} \tag{2}$$

Where the following symbols stand for:

ψ_1 - relative increment of the steam pressure in the steam volume,
φ - relative increment of the turbine rotor angular velocity,
$T_{\psi 1}$ - time constant of the turbine rotor,
T_φ - time constant of the turbine rotor,
R_μ - time constant of the steam volume,
$R_{\psi 1}$ - time constant of the steam volume,
ψ_0 - relative increment of the steam pressure before the manoeuvring valve,
$R_{\psi 0}$ - time constant of the turbine rotor,
μ - relative change of the position of the manoeuvring valve,
ψ_2 - relative increment of the steam pressure in the main condenser,
$T_{\psi 2}$ - time constant of the boiler.

1.2. System dynamic mental-verbal model of marine steam turbine

On the basis of a mathematical model, or the explicit form of the mode equation of the marine steam turbine (1) it is possible to determine the mental-verbal model of the marine steam turbine:

- If the relative increment of the steam pressure in the turbine steam volume ψ_1 increases the speed of the relative increment of the steam pressure in the turbine steam volume ψ_1 will decrease, which gives a negative cause-effect link (-).

- If the relative increment of the steam pressure before the manoeuvring valve ψ_o increases the speed of the relative increment of the steam pressure in the turbine steam volume will increase, which gives a positive cause-effect link (+).

- If the relative change of the position of the manoeuvring valve μ increases the speed of the relative increment of the steam pressure in the turbine steam volume will increase, which gives a positive cause-effect link (+).

- If the time constant of the steam volume R_μ increases the speed of the relative increment of the steam pressure in the turbine steam volume will decrease, which gives a negative cause-effect link (-).

- If the time constant of the turbine rotor $R_{\mu 0}$ increases the speed of the relative increment of the steam pressure in the turbine steam volume will decrease, which gives a negative cause-effect link (-).

- If the time constant of the steam volume $R_{\mu 1}$ increases the speed of the relative increment of the steam pressure in the turbine steam volume will increase, which gives a positive cause-effect link (+).

On the basis of the mathematical model, or the explicit form of the mode equation of the marine steam turbine (2) it is possible to determine the mental-verbal model of marine steam turbine:

- If the relative increment of the steam pressure in the steam volume ψ_1 increases the speed of the relative increment of the turbine rotor angular velocity will increase, which gives a positive cause-effect link (+).

- If the relative increment of the turbine rotor angular velocity φ increases the speed of the relative increment of the turbine rotor angular velocity will decrease, which gives a negative cause-effect link (-).
- If the relative increment of the steam pressure in the main condenser ψ_2 increases the speed of the relative increment of the turbine rotor angular velocity will decrease, which gives a negative cause-effect link (-).
- If the time constant of the turbine rotor $T_{\psi 1}$ increases the speed of the relative increment of the turbine rotor angular velocity will decrease, which gives a negative cause-effect link (-).
- If the time constant of the turbine rotor T_{φ} increases the speed of the relative increment of the turbine rotor angular velocity will increase, which gives a positive cause-effect link (+).
- If the time constant of the turbine rotor $T\psi_1$ increases the speed of the relative increment of the turbine rotor angular velocity will decrease, which gives a negative cause-effect link (-).
- If the time constant of the turbine rotor $T_{\psi 2}$ increases the speed of the relative increment of the turbine rotor angular velocity will increase, which gives a positive cause-effect link (+).

1.3. System dynamic structural model of the marine steam turbine

On the basis of the stated mental-verbal models it is possible to produce structural diagrams of the marine steam turbine, as shown in Figures 2, 3 and 4.

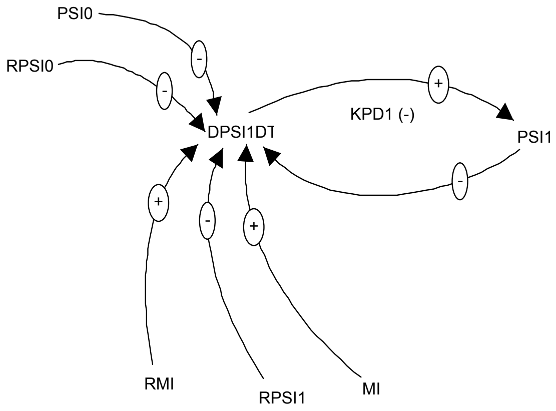


Figure 2: Structural Model of the Steam Turbine Steam Volume

In the observed system there is the feedback loop (KPD1).

KPD1(-):PSI1=>(-)DPSI1DT=>(+)DPSI1DT=> (+)PSI1; which has self-regulating dynamic character (-), because the sum of negative signs is an odd number.

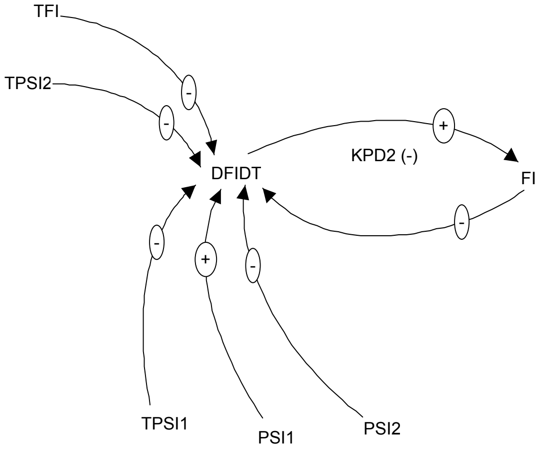


Figure 3: Structural Model of the Marine Steam Turbine – Rotor Dynamics

In the observed system there is the feedback loop (KPD2).

KPD2(-):FI=>(-)DFIDT=>(+)DFIDT=>(+)FI; which has self-regulating dynamic character (-), because the sum of negative signs is an odd number.

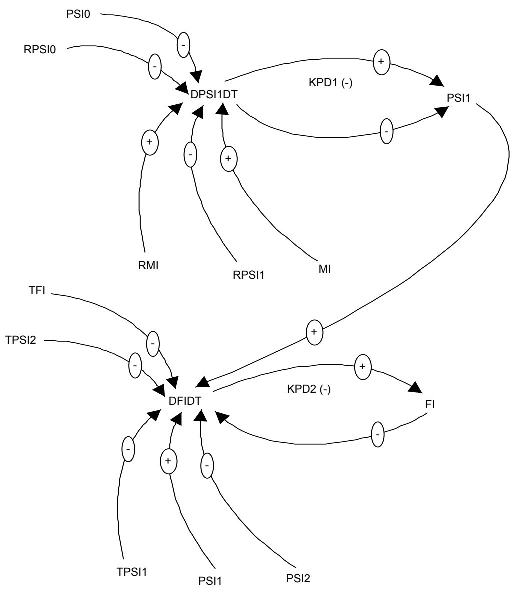


Figure 4: Global and Structural Model of the Marine Steam Turbine

1.4. System dynamic flowcharts of the marine steam turbine

Flowcharts shown in Figures 5, 6 and 7 are based on the produced mental-verbal and structural models.

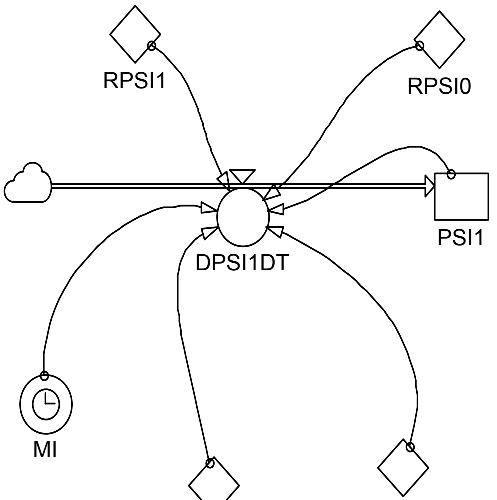


Figure 5: Marine Steam Turbine Flowchart – Steam Volume

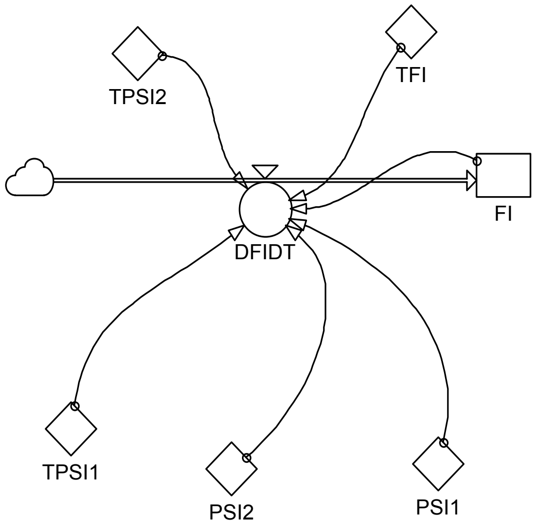


Figure 6: Marine Steam Turbine Flowchart – Rotor Dynamics

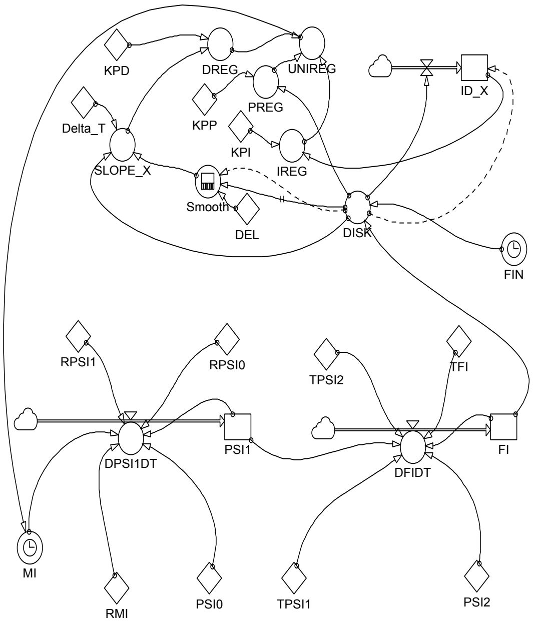


Figure 7: Global Flowchart of the Marine Steam Turbine with built-in PID Governor

MACRO DYNAMO functions built in the simulation model of the marine steam turbine: CLIP, STEP, UNIREG

2. QUANTITATIVE SIMULATION MODEL OF THE MARINE STEAM TURBINE

Simulation model of the marine steam turbine in the DYNAMO simulation language:

```
MACRO SLOPE(X, DEL)
*
A SLOPE.K=(X.K-SMOOTH(X.K,DEL))/DT
*
MEND
* ................................
* UNIREG-PID REGULATOR:
*
MACRO UNIREG(X, KPP, KPI, KPD)
*
INTRN IBD, PREG, IREG, DREG
*
A PREG.K=KPP*X.K
*
L IBD.K=IBD.J+DT*X.J
*
N IBD=X
*
A IREG.K=KPI*IBD.K
*
A DREG.K=KPD*SLOPE (X.K, DT)
*
A UNIREG.K=PREG.K+IREG.K+DREG.K
*
MEND
*
R DPSI1DT.KL=(MI.K/RMI.K)+
(PSIO.K/RPSIO.K)-(PSI1.K/RPSI1.K)
*
L PSI1.K=PSI1.J+DT*DPSI1DT.JK
*
N PSI1=0
*
A MI.K=CLIP(STEP(.05,10)+STEP(.95,50)+
PIDFI.K,0,DELAY1(RE.K,2),1E-16)
*
A RMI.K=5
*
A PSIO.K=0
*
A RPSIO.K=5
*
A RPSI1.K=5
*
SAVE DPSI1DT, PSI1, MI, RMI, PSIO, RPSIO, RPSI1
*
R DFIDT.KL=(PSI1.K/TPSI1.K)-
(PSI2.K/TPSI2.K)-(FI.K/TFI.K)
*
L FI.K=FI.J+DT*DFIDT.JK
*
N FI=0
*
A TPSI1.K=5
*
A PSI2.K=0
*
A TPSI2.K=5
*
A TFI.K=.1+MEL.K
*
* UNIREG-PID REGULATOR INSTALLING:
*
A DISK.K=FIN.K-FI.K
*
A FIN.K=STEP (.05, 10) +STEP (.95, 50)
*
A PIDFI.K=CLIP (UNIREG (DISK.K, KPP, KPI,
KPD), 0, TIME.K, 10)
*
C KPP=100
*
C KPI=0.1
*
C KPD=100
SAVE DISK, PIDFI, FIN
*
SAVE TPSI1, PSI2, TPSI2, FI, TFI
```

3. INVESTIGATING PERFORMANCE DYNAMICS OF THE MARINE STEAM TURBINE IN LOAD CONDITIONS

After system dynamics qualitative and quantitative simulation models were produced, all possible operating modes of the system will be simulated in a laboratory, using one of the simulation packages, most frequently DYNAMO (Richardson and Aleksander 1981) or POWERSIM (Byrknes 1993).

After the engineer, designer or a student have conducted a sufficient number of experiments, or scenarios, and an insight has been obtained about the performance dynamics of the system using the method of heuristic optimisation, optimisation of any parameters in the system may be performed, provided that the model is valid.

In the presented scenario the two phases of the momentum (starting) of the marine steam turbine will be presented, as well as connecting the marine synchronous generator in TIME = 100 seconds in the following way:

1. The manoeuvring valve of the marine steam opens for 5% of the rated opening in TIME = 10 seconds. The lower RPM is maintained for 50 seconds (about 5% of the rated RPM or 500-600/min.) for even heating of turbine masses.
2. In TIME = 50 seconds the manoeuvring (governing) valve opens to the rated opening (100%) MI=STEP (.05, 10) +STEP (.95, 50) and increases the marine steam turbine to the rated RPM.

   In TIME = 10 seconds the relative increment of the steam pressure in steam volume is increasing (PSI1) and also the relative increment of the angular speed of the marine steam turbine rotor (FI).

3. In TIME = 100 seconds a step load is made from 50% of the rated load, the same as in the previous scenario, and by adding stochastic load:
   TFI.K=STEP (2.5,100)*(1-NOISE())
4. Electronic PID governor has been installed, of parameters: KPP = 100, KPI= .1 and KPD = 100.

Graphic presentation of the simulation results:

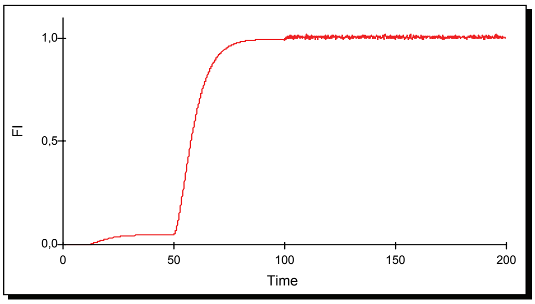


Figure 8: Relative Increment of the Angular Speed of the Rotor FI

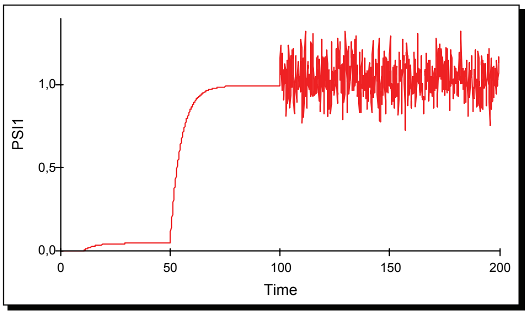


Figure 9: Relative Increment of the Steam Pressure in the Steam Volume PSI1

The results of the simulation show the real performance dynamics of the marine steam turbine, which at idle speed starts in at least two stages, and which gives sufficient time for all the parts to heat equally. This scenario may be used in heuristic optimisation of the PID governor coefficient. In fact, if the allowed criteria are reached, then in normal operating conditions the selected combination of PID governor will certainly be satisfactory. The scenario shows that when selecting the coefficient of the universal PID governor (KPP = 100, KPI = .1, KPD = 100), it will soon lead to stabilisation of the transition phase, within the limits of the rated speed deviation of the marine steam turbine rotor (approx. 4% of the rated RPM).

4. CONCLUSION

System dynamics is a scientific method which allows simulation of the most complex systems. The method used in the presented example demonstrates a high quality of simulations of complex dynamic systems, and provides an opportunity to all interested students or engineers to apply the same method for modelling, optimising and simulating any scenario of the existing elements.

Furthermore, the users of this method of simulating continuous models in digital computers have an opportunity to acquire new information in dynamic system performance. The method is also important because it does not only refer to computer modelling, but also clearly determines mental, structural and mathematical modelling of the elements of the system.

This brief presentation gives to an expert all the necessary data and the opportunity to collect information about the system in fast and scientific method of investigation of a complex system.

REFERENCES

Byrknes, A. H.,1993. *Run-Time User's Guide and Reference Manual*. Powersim 2.5, Powersim Corporation, Powersim AS, 12007 Sunrise Valley Drive, Reston: Virginia 22091 USA.

Forrester, Jay W.,1973/1971. *Principles of Systems*. Cambridge: Massachusetts USA.

Hind, A.,1968. *Automation in merchant marines*. London.

Isakov, L.I. and Kutljin, L.I.,1984. *Kompleksnaja avtomatizacija sudovljlh dizeljnih i gazoturbinmljih ustanovok*. Leningrad: Sudostreonnie.

Munitic, A.,1989. Computer *Simulation with Help of System Dynamics*. Croatia: BIS Split.

Nalepin,R.A.and Demeenko,O.P.,1975. *Avtomatizacija sudovljih energetskih ustanovok*. Leningrad: Sudostroennie.

Richardson, G. P. and Aleksander L.,1981. *Introduction to System Dynamics Modelling with Dynamo*. MIT Press, Cambridge: Massachusetts USA.

Suprun,G.F.,1972. *Sintezsistem elektroenergetiki sudov*. Leningrad: Sudostroenie.

Tireli, E., Dvornik, J., 2006. Simulation Modelling Perfomance Dynamics of Ship Gas Turbine at the Load of the Ship Synchronous Generator. *4th International Workshop on Modeling, Simulatior, Verification and Validation of Enterprise Information Systems, MSVVEIS 2006,* pp. 157-161.May 23-24, Paphos (Cyprus).

ASSESSING POOLING POLICIES IN MULTI-RETAILER INVENTORY SYSTEM WITH LOST SALES

Mounira Tlili [(a)], Mohamed Moalla [(a)], Jean-Pierre Campagne [(b)], Zied Bahroun[(a)]

[(a)]LIP2, Faculté des Sciences de Tunis, 2092 Manar 2, Tunis, Tunisie
[(b)] LIESP, INSA de Lyon 19, Avenue Jean Capelle, 69621 Villeurbanne Cedex, France

[(a)]mounira.tlili@fst.rnu.tn, mohamed.moalla@fst.rnu.tn, zied.bahroun@fst.rnu.tn [(b)]jean-pierre.campagne.insa-lyon.fr

ABSTRACT

In this paper, we address a multi-retailer inventory system with transshipment. Transshipment is used as recourse action occurring after an expected demand is realized. The remaining unsatisfied demand after transshipment is lost. We use various pooling policies for transshipment. We also develop a simulation model for such system that allows us to characterize the effects of pooling policies with/without constraints on system performance measures. The constraints are system lost sales per period and capacity per tranship.

Keywords: multi-retailer inventory system, transshipment, lost sales

1. INTRODUCTION

Collaboration between locations is a big challenge until today in a multi-echelon, multi-location inventory system. The lateral transshipment between locations at the same echelon is the most popular method of collaboration. The lateral transshipment means that the locations share the stocks (or stock pooling.) There are two approaches to share the stock. The first, if a location cannot satisfy an actual demand, other location(s) with stock on hand may ship stock to the location where the demand occurred. It is called emergency transshipment. The second, transshipment occurs as redistribution of stock before the realization of the demand, so it is called preventive transshipment. With both lateral transshipments, we can reduce costs and improve service level even if the same total stock still maintain. In general, it is assumed that the lead-time for transshipments is zero.

In multi-echelon systems, it is often assumed that customer demand appears only at the lowest echelon. The most common assumption is that the customer demand follows stationary stochastic process. Moreover, if a customer places an order it is often assumed that the customer will wait until the order arrives. This means that backorders are allowed in the model. However, if the market is competitive a lost sales model should be used. We shall also distinguish continuous review from periodic, since these models are quite different in the analysis.

The remainder of the paper is organized as follows. In section 2, we review the relevant literature in the transshipment domain. In section 3, we describe the multi-retailer transshipment problem and state the optimization problem. In section 4, after reviewing the classical expressions for computing the reorder point s and the order-up-to level S in single location inventory system, we present the simulation model. We then, propose a methodology to establish a simulation-optimization model for finding s and S that minimize the total cost. In section 5, we evaluate the benefits pooling policies over a wide range of parameter values. Finally, in section 6, we present our conclusion and some future researches.

2. THE TRANSSHIPMENT LITERATURE

Since the literature on transshipment problem is very large, we limit, in this section, to provide an overview of some works that have addressed (i) the emergency transshipment and (ii) the difficulties encountered in transshipment inventory system.

Robinson (1990) has formulated a multi-period (finite horizon), multi-location problem with emergency transshipment. In a backordered model, Robinson has proven the optimality of the order-up-to policy under the assumptions of instantaneous replenishment and transshipment lead-times. In fact, the order-up-to levels can be found analytically only when there are two retailers or the cost parameters are identical at all retailers. He has proposed a heuristic for the general case. Tagaras and Cohen (1992) have considered an inventory system consisting of a central warehouse and two retailers. They have assumed non-negligible replenishment lead-times and instantaneous transshipment times. They have allowed transshipment not only in cases of stock-outs but in also when the inventory position at a retailer falls below a critical level. The authors have defined four pooling policies that determine when and how much stock is transshipped from one retailer to the other; two policies are based on inventory level and the two other policies are based on inventory position. They have used simulation and grid-search to identify the optimal order-up-to levels. They have compared expected costs of the four policies by simulation and find that the best policy is complete pooling (the transshipped quantity is equal to the minimum of the surplus and the shortage.) Their analysis is extended by Tagaras (1999) to three retailers with identical costs. He has pointed out that the transshipment policy complicates significantly the problem and has investigated the random, risk balancing, and priority policies using optimization by simulation.

The continuous review policies are often applied in connection with spare parts (or repairable items.) Models for spare parts are commonly used in military application, and the relative literature is quite extensive. These models often assume that spare parts are characterized by high costs and low demand (Poisson process.) The one-for-one replenishments, i.e. (S-1, S) policy, is the most popular policy. The METRIC model (Multi Echelon Technique for Recoverable Item Control) is commonly used as a basic model (Sherbrooke 1968). Sherbrooke has approximated the real stochastic lead-time by its mean. Lee (1987) has extended the Sherbrooke's model by allowing lateral transshipment. In the case of identical locations, Lee has proposed three different lateral transshipment sourcing rules (random source rule and two priority rules.) Axsäter (1990) has extended the Lee's model by allowing non-identical locations.

The (s, Q) continuous review system is also used by some authors. In the inventory system with lost sales, Needham and Evers (1998) have shown that the penalty cost is the primary determinant on transshipment benefits. Evers (2001) has considered "all or nothing" policy (i.e., satisfy all the remaining demand or no) and variable transshipment cost per unit transshipped. In order to solve this problem, Evers has developed heuristic approaches to determine when transshipments should be made. Xu, Evers and Fu (2003) have dealt with the emergency transshipment in a multi-retailer inventory system with backorders. They have introduced to the classic (s, Q) policy a third parameter, hold-back level (H), which controls the level of outgoing transshipments. That is, if a retailer has only a few units on hand, it may choose not to share its inventory with the stocked-out retailer.

The literature relating to our model is not extensive. Hu, Watson and Schneider (2005) have examined a periodic review (s, S) policy in a multi-retailer system with centralized ordering and demand backordered. They have assumed that the lead-times of replenishment and transshipment are negligible, and the cost parameters at retailers are identical. Under these restrictive hypotheses, they have established a dynamic programming model in order to find the approximate optimal (s, S) policy of the system instead of each retailer. Kurkreja and Schmidt (2005) have considered a continuous review (s, S) policy in a multi-retailer inventory system with demand backordered. They have dealt with an optimization problem to obtain (s, S) policies for each retailer that minimize the total cost and satisfy the service requirement. In addition, they have proposed a research procedure based on a simulation methodology in which only the values of s are manipulated; S is obtained by adding the economic order quantity to s.

In both (s, Q) and (s, S) inventory systems, the commonly hypothesis used in all mentioned models is to consider at most one outstanding order per replenishment cycle. That is, no order arrives at the reordering period, i.e. Q>s or S-s>s.

In the literature, the analytical models can only be found under restrictive assumptions (instantaneous replenishment, two locations, and identical costs). In fact, relaxing the assumption of instantaneous replenishment complicates significantly the mathematical analysis, because of the interrelationships among demand, transshipment quantities and pipeline inventories, and consequently the state space should be expanded (Tagaras 1999). In addition, Tagaras (1999) has shown that the exact model becomes intractable even in the case of two locations. Minner, Silver and Robb (2003) have also indicated two other reasons: (i) transshipments have a secondary effect in changing the time and the size of replenishment orders at a location releasing the transshipment, and (ii) the establishment of safety stocks must take into account the transshipment possibilities instead of the constraint of desired service levels. For these reasons, many authors have resorted to simulation in order to study multi-retailer inventory system under relaxed hypothesis. It is in this direction that our work is pointing out.

We are interested to study a periodic review (s, S) with lost sales in a multi-retailer inventory system integrating transshipment. The simulation model is designed to search policy parameters at each retailer that reduce the total system cost (holding, ordering, penalty, and transshipment costs). In a recent work, Tlili *et al.* (2008) have examined the benefits of complete pooling and all or nothing policies in a two-retailer inventory system and have concerned to study the effects of transshipment and penalty costs. Tlili *et al.* have evaluated these transshipment policies under identical/non-identical replenishment lead-times. For this system, the authors have proposed an effective procedure based on a grid-search by coupling simulation and optimization. This procedure is very appealing to practitioners in inventory management. The most important findings can be summarized in: (i) the type of transshipment policies does not have a significant difference on the system performance measures, (ii) the transshipment is more efficient only when the transshipment cost is non expensive in comparison with the penalty cost, and (iii) it is preferable to design distribution systems with identical replenishment lead-times and demand parameters in order to achieve a desirable savings of transshipments. In this paper, we relax some assumptions and extend the results of this earlier study to more than two retailers. In addition, our contribution in this work is threefold. First, we study the transshipment problem using complete pooling under with/without the hypothesis that at most one outstanding order is possible per replenishment cycle. Second, we also examine these two systems by adding lost sale constraints per period at system level (all retailers). Third, we deal with a constraint of capacity per tranship instead of hold-back levels at retailers. Next, we present our transshipment problem in a multi-retailer inventory system.

3. MULTI-RETAILER TRANSSHIPMENT PROBLEM

3.1. Description of the problem

We study an inventory pooling system of multi-retailers, which are replenished by a warehouse. Each retailer i allows a periodic review (s_i, S_i) ordering policy and faces random demand, which is normally distributed (mean μ_i and standard deviation σ_i) and independent of the demand at the other retailers. We assume that the warehouse does not keep any stock; i.e., any ordering policy is adopted by the warehouse. The replenishment lead-time from the warehouse to retailers is L periods and a fixed cost (K) is associated with each replenishment order. If retailer i places an order at period t, it will arrive at the beginning of period t+L+1. The emergency transshipment is allowed from retailers having excess stock to the retailers having shortage stock with zero time of transshipment. The quantity transshipment from j to i in period t is denoted by $X_{j,i,t}$ and a fixed transshipment cost (ct) is associated with this activity. The transshipment cost is independent of the number of units transshipped as well as the number of transshipment requests. This can be justified only when all units to be transshipped from one retailer to another can be transshipped by a single shipment. If the transshipment is not possible or the demand is partially satisfied via transshipment, the remaining unmet demand is lost and a penalty cost (cp) per unit lost occurs. Furthermore, at the end of the review period, the remaining stock is subject to holding cost (ch) per unit and period unit.

We define a system as a group of retailers. The α-service (no stock-out probability) and β-service (fraction of satisfied demand) levels are measured at the system level.

3.2. Notation

N	Number of retailers
$\overline{\beta}$	Pre-assigned fraction for the number of lost sales per period at the system level
DS	Desired system service level
Q_{max}	Capacity maximum per tranship
k_i	Safety factor for retailer i
$x_{i,t}$	Demand at retailer i in period t
$y_{i,t}$	Inventory level at retailer i in period t
$G_u(k_i)$	The unit normal function (mean 0, standard deviation 1)
$\delta_{i,t}$	1 if retailer i in period t is still in a shortage situation and 0 otherwise
$TX_{i,t}$	Total transshipment quantity towards retailer i in period t
EOQ	Economic order quantity

$$EOQ = \sqrt{\frac{2K\mu_i}{ch}} \tag{1}$$

3.3. Transshipment policies

3.3.1. Complete pooling

In a multi-retailer system, if one retailer could not satisfy its local demand from its own inventory on hand, it could be place a transshipment request to all the other retailers which are capable of providing such excess stock. Thus, the total transshipment quantity into retailer i from the other retailers is:

$$TX_{i,t} = \sum_{j=1, j\neq i}^{N} X_{j,i,t} * \delta_{i,t} = \min\left\{[x_{i,t} - y_{i,t}]^{+}, y_{j,t} * \delta_{i,t}\right\} \tag{2}$$

Where $[x]^{+} = \max(0, x)$

The principal rule of complete pooling is to satisfy the unmet demand as long as there are available stocks at the other retailers and there is still shortage at retailer i ($\delta_{i,t}$=1).

3.3.2. Partial pooling

We consider a partial pooling policy that uses a capacity constraint per tranship (Q_{max}). In this situation, if a retailer has stocked out upon the unexpected of a customer demand, it places a transshipment request to the other (requested) retailer. The requested retailer is allowed to tranship any inventory lower than the quantity maximum (Q_{max}) to be transshipped. The capacity constraint can be interpreted as the capacity of the cargo that can be transported. Thus, the partial pooling policy under capacity constraint for multi retailers can be formulated as below:

$$\begin{aligned} &if \;\; x_{i,t} - y_{i,t} \leq \sum_{j=1, j\neq i}^{N} y_{j,t} * \delta_{i,t} \;\; then \;\; X_{j,i,t} = \min\left([x_{i,t} - y_{i,t}]^{+}, Q_{\max}\right) \\ &else \;\; X_{j,i,t} = \min\left(\sum_{j=1, j\neq i}^{N} y_{j,t} * \delta_{i,t}, Q_{\max}\right) \end{aligned} \tag{3}$$

3.3.3. Sequential procedure of transshipment

For both complete pooling and partial pooling policies, we should specify the transshipment decisions. For simplicity, we consider a sequential procedure to determine from which retailer the transshipment will be requested first. For instance, in the case of N=4, the retailers are numbered 1, 2, 3 and 4. While retailer 1 is in shortage situation at period t (i.e., $\delta_{1,t}$=1), the transshipments follow the sequence of retailers 2, 3 and 4. Nevertheless, transshipments depend on the available stock at these retailers:

Condition 1: if retailer 2 has sufficient stock to fill the requested quantity hence the transshipment is realized, so $\delta_{1,t}$=0.

Condition 2: if retailer 2 has an available stock but it is lower than the requested quantity then we satisfy partially the request, so $\delta_{1,t}$=1. The request corresponding to the remaining quantity is transmitted to retailer 3. If there is still shortage at retailer 1 ($\delta_{1,t}=1$), the transshipment request is sent to retailer 4.

Condition 3: if retailer 2 is in shortage situation the transshipments follow the sequence of retailers 3, 4 and 1. Hence, retailer 1 will be satisfied from retailer 3 and 4 after the transshipment decisions of retailer 2.

It is worth noting that retailers 3 and 4 should also verify the conditions 1, 2 and 3.

3.4. The optimization problem

We are concerned with finding (s, S) policies for each retailer that minimize the total system cost E(TC). The total system cost consists of ordering, holding, penalty, and transshipment costs. We want to study on assessing the benefits of pooling policies (complete and partial pooling) under with/without the hypothesis that at most one outstanding order per replenishment cycle (S-s>s) and under the system lost sales constraint per period ($\overline{\beta}$). Thus, we have the following models:

Model 1: Complete pooling without S-s>s constraint

$$\begin{aligned} &\underset{(s_i,S_i);\, i=1\ldots N}{Min} \quad E(TC) \\ &s.t. \quad s_i \geq 0, S_i \geq 0, \text{ and Integers}; i = 1\ldots N \end{aligned}$$

Model 2: Complete pooling with S-s>s constraint

$$\begin{aligned} &\underset{(s_i,S_i);\, i=1\ldots N}{Min} \quad E(TC) \\ &s.t. \quad s_i \geq 0, S_i \geq 0, \text{ and Integers}; i = 1\ldots N \\ &\qquad S_i - s_i \geq s_i;\ i = 1\ldots N \end{aligned}$$

Model 3: Complete pooling without S-s>s constraint

$$\begin{aligned} &\underset{(s_i,S_i);\, i=1\ldots N}{Min} \quad E(TC) \\ &s.t. \quad s_i \geq 0, S_i \geq 0, \text{ and Integers}; i = 1\ldots N \\ &\qquad \text{system lost sales per period} \leq \overline{\beta} \end{aligned}$$

Model 4: Complete pooling with S-s>s constraint

$$\begin{aligned} &\underset{(s_i,S_i);\, i=1\ldots N}{Min} \quad E(TC) \\ &s.t. \quad s_i \geq 0, S_i \geq 0, \text{ and Integers}; i = 1\ldots N \\ &\qquad S_i - s_i \geq s_i;\ i = 1\ldots N \\ &\qquad \text{system lost sales per period} \leq \overline{\beta} \end{aligned}$$

Model 5: Partial pooling without S-s>s constraint

$$\begin{aligned} &\underset{(s_i,S_i);\, i=1\ldots N}{Min} \quad E(TC) \\ &s.t. \quad s_i \geq 0, S_i \geq 0, \text{ and Integers}; i = 1\ldots N \\ &\qquad X_{j,i,t} \leq Q_{\max};\ i, j = 1\ldots N \text{ and } j \neq i \end{aligned}$$

Model 6: Partial pooling with S-s>s constraint

$$\begin{aligned} &\underset{(s_i,S_i);\, i=1\ldots N}{Min} \quad E(TC) \\ &s.t. \quad s_i \geq 0, S_i \geq 0, \text{ and Integers}; i = 1\ldots N \\ &\qquad S_i - s_i \geq s_i;\ i = 1\ldots N \\ &\qquad X_{j,i,t} \leq Q_{\max};\ i, j = 1\ldots N \text{ and } j \neq i \end{aligned}$$

In fact, our transshipment problem with lost sales is a complex problem and hard to solve it with analytic method, so we have recourse to solve this optimization problem by integrating the simulation with the optimization.

4. RESOLUTION METHOD

4.1. Initial phase

Before we begin the simulation, we should have an initial phase. In the inventory literature, many research works are interested to determine the policy parameters under various assumptions for a single location/single product inventory system. Since our model deals with a normal distribution demand, an exact approximation of the reorder point based on service level constraint is derived by Schneider and Ringuest (1990). Let (s_i^0, S_i^0) denotes the initial values for each retailer i.

$$s_i^0 = \mu_i(L+1) + k_i \sigma_i \sqrt{L+1} \tag{4}$$

In lost sales systems, the safety factor k_i is a solution of the equation (5). The numerical results are given in (Silver, Pyke and Peterson, 1998) p. 725-734.

$$G_u(k_i) = \frac{EOQ}{\sigma_i \sqrt{L+1}} \left(\frac{1-DS}{DS} \right) \tag{5}$$

Finally, the order-up-to-level is determined as:

$$S_i^0 = s_i^0 + \text{EOQ} \tag{6}$$

Wagner, O'hagan and Lundh (1965) have indicated that the economic order quantity Q is a good approximation for the optimal reorder quantity under the condition that the ratio K/ch is relatively high to μ_i, which is the case in our study.

4.2. Simulation-optimization model

The simulation-optimization models corresponding to the six models (section 3.4) are implemented on a spreadsheet and run by both Monte Carlo simulation and OptQuest of Crystal Ball 7.2®.

The simulation model is operated on a basic time period of one period; i.e., the inventory is reviewed once each period. The demand per period is considered as the input of the model. We use Monte Carlo techniques to generate the demands. The rule of Monte Carlo simulation is to select randomly the customer demands according a normal distribution, which is specified by a mean and a standard deviation. In order to confirm the hypothesis of i.i.d distribution we should disable the correlation between demands over time and among retailers. As output, the simulation model saves a variety of system performance measures: the total cost, the system service levels (α-service and β-service), transshipment rate (total transshipment quantity/total demand), and lost sale rate (total lost sales/total demand). The simulation model is validated by using a 95% confidence interval for all the performance measures. In addition, the simulation model is run over a planning horizon of 100 period units and each period is simulated 1000 times, so that the system performance measures are more accurate; i.e., low mean standard error and low coefficient of variability.

The optimization phase is based on a wide grid-search for (s_i, S_i), i=1…N. In order to construct this grid, we set the values of s_i and S_i within two given intervals; $s_i \in [LB_{1,i}, UB_{1,i}]$ and $S_i \in [LB_{2,i}, UB_{2,i}]$. The bounds of these intervals are chosen based on the following idea:

- The lower bounds for (s_i, S_i) are computed as follows: $LB_{1,i}$ is equal to demand during L periods and $LB_{2,i}$ is equal to demand during the replenishment cycle (L+1 periods);
- The upper bounds for (s_i, S_i) are the initial values of the reorder points and the order-up-to levels (determined in section 3.3).

Since the minimization criterion is not a convex function for the (s, S) inventory system; i.e. present several distinct relative minima (Wagner, O'hagan and Lundh 1965), we have to search the best combination (s_i, S_i), i=1…N, through this grid. However, the number of all possible combinations within this grid is combinatory. So, we run the OptQuest optimization tool of Crystal Ball 7.2® by using two different step sizes (five units and one unit). For each step size, we run OptQuest for 1000 simulations.

By starting with the initial solution (s_i^0, S_i^0), the research for (s_i, S_i) is carried out in $[LB_{1,i}, UB_{1,i}]$ and $[LB_{2,i}, UB_{2,i}]$ with a step size of five units. At the end of this phase, OptQuest of Crystal Ball provides the best solution among all the evaluated solutions which minimize the total cost and satisfy the constraints. We denote this solution by (s'_i, S'_i) for each retailer i, then, a new research is made closely around this solution by using a one step size and restricting the intervals as below:

$LB_{1,i} = s'_i-5$; $UB_{1,i} = s'_i+5$
$LB_{2,i}= S'_i-5$; $UB_{2,i}= S'_i -5$

Finally, we obtain the best values (s_i, S_i) for each retailer i achieving our objective.

We firstly run the simulation-optimization model for one retailer by using the initial values s_i^0 and S_i^0 as starting points. This phase gives us the optimal inventory control parameters (s, S) for the no pooling system. These optimal values will be served as starting points for the multi-retailer pooling inventory system.

5. NUMERICAL RESULTS

The numerical analysis are reported to show the sensitivity of the constraints made upon the mean lost sales and capacity per tranship on the performance measures. The experiments are evaluated via large combination of the input parameters: DS=98%, K=100, ch=1, cp=100, ct=20, EOQ=89, the other parameters are given in Table 1 and the initial solutions for the corresponding parameters are presented in Table 2. According to this combination, we have in total 360 problems to be evaluated.

Table 1: Input Parameters for Numerical Examples

Parameter	Levels	Values
N	3	2; 4; 8
L	2	1; 3
σ_i	2	10; 20
$\bar{\beta}$	4	0.1; 0.05; 0.02; 0.01
Q_{max}	3	5; 10; 15

Table 2: Initial Solutions

L	μ_i, σ_i	k_i	s_i^0	S_i^0
1	40, 20	1,13	112	201
3	40, 20	1,3	212	301
1	40, 10	0,76	91	180
3	40, 10	0,95	179	268

5.1. Evaluation of models 1 and 2

An examination of Table 3 leads to several interesting observations. In all cases, the total cost in model 2 is higher than that in model 1. This can be explained by the increase/decrease of both lost sales and transshipment rates (indicate by %LS and %TR in Table 3.) We can also observe that the difference between these two models declines with the number of retailers and becomes not significant (lower than 1%) in the case of shorter replenishment lead-time (L=1) and large number of retailers. However, the α-service in model 2 is lower than that in model 1 and the difference between two models goes up 2.11%. This can be interpreted by the increase of lost sale occurrences in the model 2 due to the severe constraint S-s>s and the partial satisfaction via transshipments. Moreover, the difference between two models in β-service is negligible and does not exceed 0.19%. This is due to the transshipments, which tend to satisfy immediately customer demand from stocks on hand at all retailers and alleviate partially or entirely the shortages.

We can also draw another conclusion, which concerns the stock levels. In most cases, the reorder points (s) in model 2 are more reduced than those in model 1 and, in all cases, the order-up-to levels (S) in model 2 are strongly higher than those in model 1. Indeed, in model 1, each retailer tends to have a higher reorder point, and so takes earlier precaution for lost sales by placing frequently orders with small quantity. While in model 2, the higher order-up-to levels are due to the constraint S-s>s which leads to place less frequently orders with large order quantity. Hence, the number of replenishment cycles in model 1 is about two times than that in model 2. That is, an order is placed in the middle of a replenishment cycle. This leads to have a higher reorder point and a lower order-up-to level and consequently the holding and penalty costs are more reduced in model 1 than that in model 2.

Table 3: Difference in Performance Measures of Model 1 vs. Model 2

N	σ_i	L	%α	%β	%E(TC)	%TR	%LS	%s	%S
2	10	1	0,00	0,02	0,91	-0,79	-20,01	0,62	14,41
	10	3	-1,55	-0,04	12,92	33,01	19,55	-5,79	26,36
	20	1	0,00	0,03	1,46	-9,50	-18,16	-1,63	10,03
	20	3	-2,11	-0,07	11,40	29,48	18,08	-9,44	21,68
4	10	1	-0,80	-0,03	0,20	11,68	19,88	-2,42	1,25
	10	3	-1,23	0,00	8,11	24,96	-2,71	-6,75	23,60
	20	1	-0,13	-0,02	0,10	5,19	9,02	-2,82	1,58
	20	3	-1,08	-0,06	5,17	38,37	23,28	-11,45	18,46
8	10	1	0,04	0,01	0,26	-2,11	-7,96	1,08	0,00
	10	3	-1,67	-0,19	5,90	17,25	71,95	-9,38	19,04
	20	1	-0,13	-0,02	0,10	5,19	9,02	-2,82	1,58
	20	3	-0,92	-0,12	1,90	23,30	47,04	-22,92	14,52

Next, we focus on accessing the benefits of transshipments according to the variation of replenishment lead-times and standard deviation of the demand for models with/without the constraint S-s>s. The cost increases with the lead-time as well as the standard deviation. For instance, in the case of two-retailer inventory system in model 2, the total cost, the transshipment rate, the reorder point, and the order-up-to level increase, respectively, in average about 19.73%, 47.59%, 46.81%, and 48.23% when the lead-time increases from one to three for σ_i=10 and these performance measures also increase, respectively, about 16.71%, 8.83%, 18.05%, and 14.18% when the standard deviation ranges from 10 to 20 for L=1. In addition, the increase of the performance measures declines with the number of retailers. That is, when the transshipment cost is non-expensive in comparison with the penalty cost, the transshipment leads to cost savings and reduction in inventories for large distribution systems. These results hold in the models 1, 3, 4, 5 and 6.

5.2. Effects of lost sales constraint

The observations that emerge from Figures 1 and 2 concerning the variation of the total cost and the transshipment rate are very interesting. For instance, in model 3, when the rate $\overline{\beta}$ becomes very tight (0.01), the total cost is lower (Figure 1). This can be explained by the decrease of the total number of lost sales due to the severe constraint $\overline{\beta}$ made upon each period. When the rate $\overline{\beta}$ is large (0.1), we obtain higher total cost even if the transshipment rate is also higher (Figure 2). Indeed, the increase of the total cost is due primary to the expensive penalty cost which dominates the transshipment cost. These results hold in the model 4.

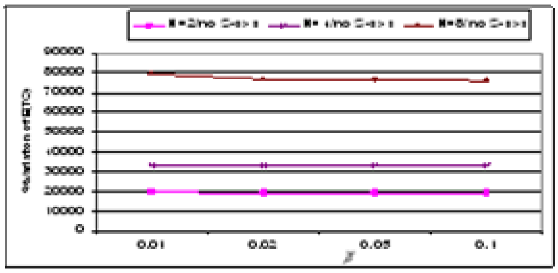


Figure 1: Variation of Total Cost for Model 3, where σ_i =10 and L= 1

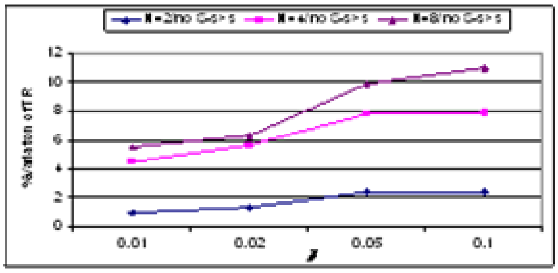


Figure 2: Variation of Transshipment Rate for Model 3 where σ_i =10 and L=1

A study of sensitivity of the constraint made upon the rate $\overline{\beta}$ indicates that the performance measures and stock levels depend on the variation of σ_i, L, and N. Figure 3 shows that, when the lead time ranges from one period to three periods, the total cost increases with the constraint $\overline{\beta}$ as well as with the number of retailers. In fact, during a longer lead time, the retailers tend to keep a large quantity of stock in order to protect against the shortage. In addition, the total cost also increases with the standard deviation (σ_i varies from 10 to 20) and the number of retailers (Figure 4). We study now the sensitivity of inventory levels to the constraint $\overline{\beta}$. In model 4, the variation of $\overline{\beta}$ from 0.01 to 0.1 leads to an increase about of 8.61% in s and about of 8.35% in S for N=2, σ_i=10 and L=3. For N=4, the increase in s is of 7.97% and in S is in average about 4.81%. For N=8, the increase in s is about 5.22% and in S is about of 3.30%. We observe that the increase of stock levels declines with the number of retailers. This can be explained by the fact that, in a large distribution system, transshipment tends to alleviate completely the shortages (i.e., the total number of lost sales is nearly zero), so the inventory levels are less sensitive to the constraint $\overline{\beta}$.

The difference between the models 3 and 4 is more significant in longer lead time and higher standard deviation. This is substantiated in Figures 3 and 4. However, the difference between these models in α-service and β-service levels does not exceed 1% in all cases. This is due to the lower effect of $\overline{\beta}$, resulting from its small value.

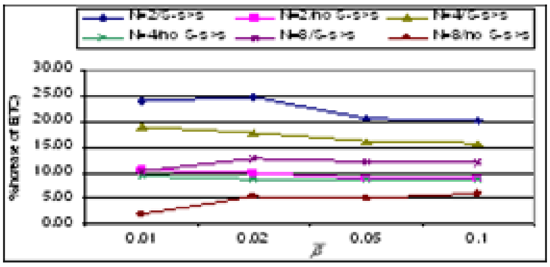


Figure 3: Sensitivity of Total Cost to the Lead Time L (from 1 to 3) where σ_i =10 for Models 3 and 4

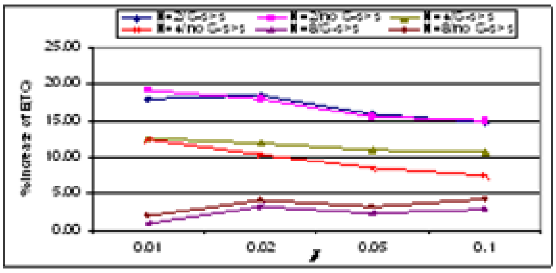


Figure 4: Sensitivity of Total Cost to the Standard Deviation σ_i (from 10 to 20) where L=1 for Models 3 and 4

5.3. Impact of capacity per tranship

We turn now our attention to examine the effects of the partial pooling on system performance measures. The degree of transshipment benefits depends on the capacity constraint per tranship. The α-service and β-service levels achieved by the models 5 and 6 are not much different in the whole range of the values. To be more specific, the system's α-service and β-service levels go higher as the capacity per tranship becomes large. Moreover, a reduction in the capacity per tranship will decline the outgoing transshipment in the retailers having stock on hand (Figure 6) and consequently will increase the lost sales (i.e., increase the penalty cost) at each retailer (Figure 5). In addition, the lower the capacity in all retailers is, the higher the stock levels at each retailer are (Figures 7 and 8). The change of the total cost due to the change of capacity per tranship appears to be as a linear behaviour (Figure 5). The models 5 and 6 show a considerable difference in the performance measures, in particular, in longer lead times. We also observe that this difference is very sensitive to σ_i, L, and N.

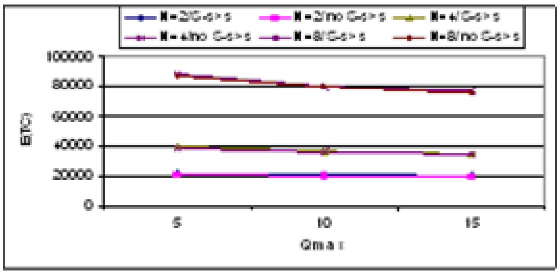


Figure 5: Variation of Total Cost for Models 5 and 6 where σ_i =10, L=1

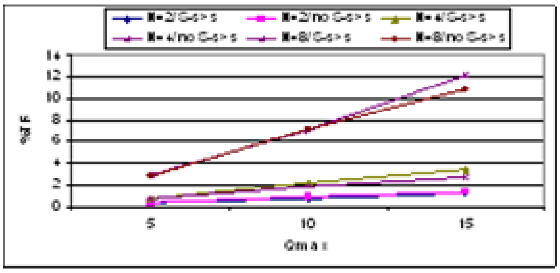


Figure 6: Variation of Transshipment Rate for Models 5 and 6 where σ_i =10, L=1

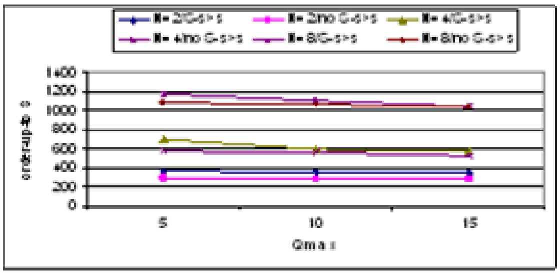


Figure 7: Variation of Order-up-to Levels in Function of Q_{max} and N where σ_i =10, L=1

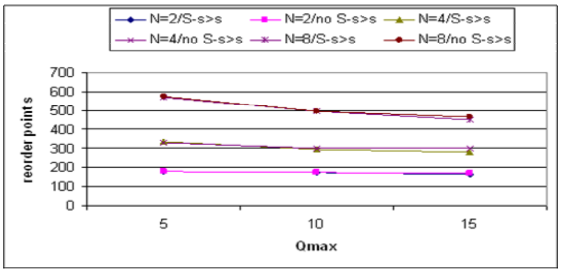


Figure 8: Variation of Reorder Points in Function of Q_{max} and N where σ_i =10, L=1

The experiments also indicate that the difference between the complete pooling and the partial pooling under large capacity (Q_{max}=15) for N=2, σ_i =10, L=1 is about 4.29% in E(TC), 3.59% in s, 5.93% in S, 0.81% in α-service, and 0.01% in β-service. Thus, for Q_{max} more than 15, the benefits with partial pooling under a large capacity per tranship tends to be similar to that of the complete pooling policy. This result holds for a large number of retailers as shown in Figures 9 and 10. In practice, a retailer would like keep some units of stock for its future needs instead of sharing all its available stock with the other retailers. In this situation, a partial pooling policy can be considered as an attractive policy to manage the stocks at retailers.

5.4. Efficiency of the EOQ

It is worth comparing the EOQ to the reorder quantity in all transshipment models. Our results indicate that the EOQ appears to be a good initial solution in models without S-s>s constraint. In all cases of models 1, 3 and 5, the reorder quantity is reduced in comparison with the EOQ. This reduction goes higher as the number of retailers becomes large and goes lower with the increase of lead-time/standard deviation. However, the EOQ is not recommended in the models with S-s>s constraint. In some cases, the reorder quantity in transshipment model is about two times than the EOQ. To be more specific, a longer lead-time and/or a higher standard deviation lead to a large gap between the EOQ and the reorder quantity in transshipment model. Nevertheless, in the model 2, the EOQ can be served as an initial solution only in the case of shorter lead-time (L=1) and lower standard deviation (σ_i=10).

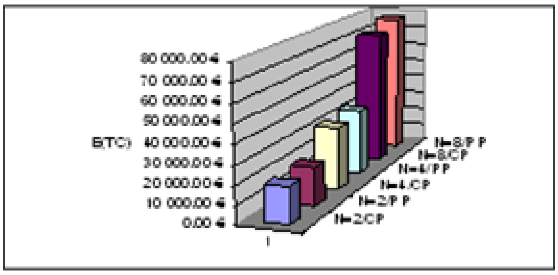


Figure 9: Complete Pooling (CP) vs. Partial Pooling (PP) where Q_{max}=15, for σ_i =10, L=1 in Model 6

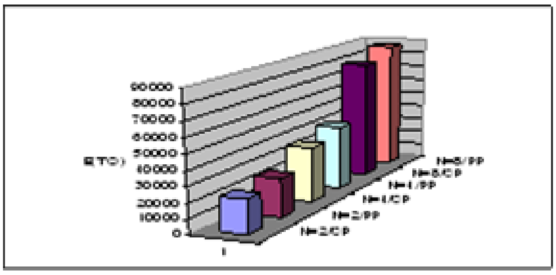


Figure 10: Complete Pooling (CP) vs. Partial Pooling (PP) where Q_{max}=15, for σ_i =20, L=1 in Model 6

6. CONCLUSION

This paper has examined the effectiveness of two transshipment policies in two-echelon distribution system, characterized by a single warehouse at the higher echelon and multiple retailers at the lower. One of these transshipment policies is a complete pooling policy, while the other is the more realistic, partial pooling policy. Our results indicate that the benefits appear to increase, as the number of retailers becomes large. Moreover, under some circumstances, either of complete or partial pooling policies, may be the most desirable policy.

Future works should focus on adding a fixed cost for each transshipment request in models 5 and 6. Another extension involves consideration of ordering policy at the warehouse and investigation of transshipment benefits.

REFERENCES

Axsäter, S., 1990. Modelling emergency lateral transshipments in inventory systems. *Management Science*, 36(11), 1329-1338.

Evers, P., 2001. Heuristics for assessing emergency transhipment. *European Journal of Operational Research*, 129, 311-316.

Hu, J., Watson E., Schneider H., 2005. Approximate solutions for multi-location inventory systems with transshipments. *Int. J. Production Economics*, 97, 31–43.

Kukreja, A., Schmidt C., 2005. A model for lumpy demand parts in a multi-location inventory system with transshipments. *Computers & Operations Research*, 32, 2059-2075.

Lee, H.L., 1987. A multi-echelon inventory model for repairable items with emergency lateral transshipments. *Management Science*, 33, 1302–1316.

Minner, S., Silver E., Robb D., 2003. An improved heuristic for deciding on emergency transhipments. *European Journal of Operational Research*, 148, 384-400.

Needham, P.M., Evers P.T., 1998. The influence of individual cost factors on the use of emergency transhipments. *Transpn Res.-E*, 34(2), 149-160.

Robinson, L.W., 1990. Optimal and approximate policies in multiperiod, multilocation inventory models with transshipments. *Operations Research*, 38 (2), 278–295.

Schneider, H., Ringuest J.L., 1990. Power approximation for computing (s; S) policies using service level."*Management Science*, 36 (7), 822–834.

Sherbrooke, C.C., 1968. METRIC: Multi Echelon Technique for Recoverable Item Control. *Management Science*, 16 (1), 122–141.

Silver, E.A., Pyke D.F., Peterson R., 1998. *Inventory management and production planning and scheduling*. John Wiley & Sons.

Tagaras, G., Cohen M., 1992. Pooling in two-location inventory systems with non-negligible replenishment lead-times. *Management Science*, 38(8), 1067–1083.

Tagaras, G., 1999. Polling in multi-location periodic inventory distribution systems. *Omega*, 27 (1), 39–59.

Tlili, M., Moalla M., Campagne J.P., Bahroun Z., 2008. Un modèle basé simulation-optimisation pour un système de distribution à deux stocks intégrant le transshipment. Proceedings 7e Conférence Internationale de MOdélisation et SIMulation (MOSIM'08), pp. 1240-1248, March 3 - April 2, Paris, France.

Xu, K., Evers P.T. and Fu M.C., 2003. Estimating customer service in a two-location continuous review inventory model with emergency transshipments. *European Journal of Operational Research*, 145(3), 569–584.

Wagner, H.M., O'hagan M. and Lundh B., 1965. An empirical study of exact and approximately optimal inventory policies. *Management Science*, 11(7), 690-723.

MODELLING AND CONTROLLER PROTOTYPING FOR UNMANNED VERTICAL TAKE OFF AND LANDING (UVTOL) VEHICLES

**Alexander Martínez[a], Pedro Gutiérrez[b], Claudio Rossi[c],
Antonio Barrientos[d], Jaime DelCerro[e], Rodrigo SanMartín[f]**

Robotics and Cybernetics Group, Universidad Politécnica de Madrid, Spain

[a]alexmartinez@etsii.upm.es, [b] pgutierrez@etsii.upm.es, [c]crossi@etsii.upm.es,
[d]antonio.barrientos@upm.es, [e]j.cerro@upm.es, [f]rsan@etsii.upm.es

ABSTRACT

This paper describes a methodology to parameterize linear, time invariant (LTI) models which represent the dynamics of UVTOLs and that are appropriate for analytical development of controllers. The models' validity was tested against real telemetry from two vehicles, a mini-helicopter and a quad-rotor. The experiments show that despite its inherent limitations the LTI models are suitable for modeling the complex dynamics of aerial vehicles. Different LTI models for the mini-helicopter's stationary, lateral and longitudinal flights were obtained. Similarly, given the geometrical and dynamic characteristics of the quad-rotor no distinction is made between stationary, lateral and longitudinal flights, and only one LTI model was obtained, which represents the overall dynamic behavior of the vehicle. Because of their relative simplicity these models were used to design analytical controllers and to obtain different controller prototypes in a quick and simple way to evaluate the UVTOL's performance in different flight conditions.

Keywords: UVTOL, modelling, controller prototyping

1. INTRODUCTION

The modeling, guidance, navigation and control of Unmanned Vertical Take Off and Landing (UVTOL) vehicles is a research topic with great developments, and has achieved significant progresses in recent years. It is sufficient to mention some work on helicopters modeling and control to note the large number of approaches that has been given to this field of engineering. As for the modeling we can cite works as the La Civita (2002), who presents a nonlinear helicopter model, derived on the basis of aerodynamic principles, and its linearization, enabling the possibility of including several equilibrium points in the flight envelope. Avila et al (2003) who develop a nonlinear model and a nonlinear control strategy for a scale model helicopter. Cunha and Silvestre (2003), introduce a helicopter dynamic simulation model specially suited for the design, test, and evaluation of flight control systems for model-scale helicopters. Castillo et al (2004) present a controller design and its implementation on a quadrotor, whose dynamic model is obtained via a Lagrange approach. DelCerro et al (2004) develop an analytical model of a small helicopter as well as a statistical study about the influence of its main parameters. The stability and response of the model is presented after a brief comparison of different techniques for helicopters dynamic modeling. Madani and Benallegue (2006) present a nonlinear dynamic model for a quadrotor helicopter in a form suited for backstepping control design. Vélez et al (2006) present a rapid software prototyping environment for the design, development and simulated test of a control system for an autonomous mini-helicopter. Amir and Abbass (2008) propose a non-linear mathematical model of quadrotor dynamics. Whit regard to control techniques used in helicopters and quadrotors, it include some as varied as robust control (Isidori 2003), backstepping (Bouabdallah and Siegwart 2005), sliding-mode (McGeoch and McGookin 2005, Rong Xu and Ozguner 2006), PID/PD and fuzzy (Castillo 2005). As you can see, there is a continued interest in applying different techniques to control this type of aircraft and also the need of modeling VTOL in each desired case.

The UVTOL's complex dynamics and its variations related to flying altitude, weather conditions, changes in the vehicle's configuration (for example: weight, payload and fuel quantity), disrupt the modeling process and, consequently, the systematic development of control systems, resulting in tedious and critical heuristic adjustment procedures. This paper proposes a modeling methodology that leads to a set of LTI models, which allow representations of stationary, lateral and longitudinal phases of flight; these models are then used for the development of analytical controllers and, as a consequence, a systematic synthesis process of them.

2. DESCRIPTION OF USED VTOL AND PROPOSED MODEL

2.1 Description of aerial vehicles

In this section we briefly describe the VTOLs used and give a description of some details about the characteristics, variables and signals involved in their flight dynamics.

VTOLs are considered systems of six degrees of freedom, defined by three degrees to the position or location (X, Y, Z) and three other degrees to attitude

(Roll, Pitch, Yaw angles) as described in Figure 1 for the mini-helicopter.

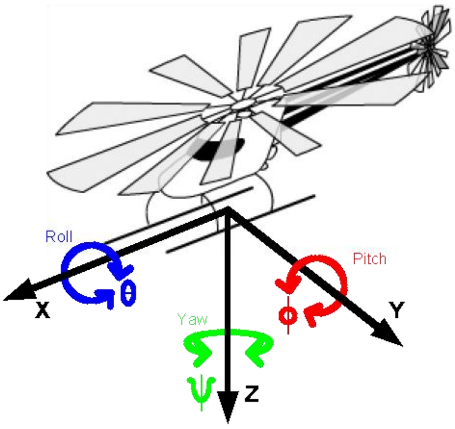


Figure 1: Nomenclature Used for the Mini-helicopter's Variables

The mini-helicopter used is a Benzin Trainer of VARIO. Its main rotor measured 1.5 meters and has a 5 kg payload, thanks to its 26 cc engine.

On the other hand, the quadrotor used is a Draganflyer SAVS, with 0.8 m diameter, a payload of 85 g and energy autonomy for 12 to 15 minutes flights.

Thanks to the instrumentation on board the aircrafts and communications systems available, data from signals involved in the flight can be stored on land. Among the variables that can be obtained, are relevant to the model developed the following signals (as graphically represented in Figure 1):

X, Y, Z: Measured respect to an inertial system and only in the case of mini-helicopter, with a GPS.

Roll, Pitch: Measured with an IMU, with respect to the mobile axis of each VTOL.

Yaw: Measured with magnetic compass in the mini-helicopter and with IMU in Quadrotor.

Vroll, Vpitch, Vyaw: Calculated using a Kalman filter from the signals measured into IMU.

Croll, Cpitch, Cyaw: Control Signals of the respective Roll, Pitch and Yaw angles, sent through of RC transmitter.

2.2 Mini-helicopter's mathematical model

Dynamic features of VTOLs can be represented by complex models, where components such as non-linearities or parameter uncertainty are not easy to determine. In contrast, the majority of controller design procedures require a relatively simple mathematical model to establish some characteristics and adjust the parameters of the controller to design.

A relatively complex mathematical model for the mini-helicopter used in this work was obtained in earlier work (Aguirre 1999, DelCerro 2007). This model offers a good input-output representation of the system, but its complexity does not allow to use it in analytical procedures for controllers design. From this model we obtained a structure of a simple mathematical model, which can be represented with linear, time invariant (LTI) state equations and use it to develop a systematic procedure of controllers design.

The mini-helicopter was considered a decoupled system respect to its three main movements: the movement in a horizontal plane, parallel to the ground, with varying angles roll, pitch and the resulting lateral and longitudinal displacement, the Yaw movement and vertical movement or Z variation.

2.2.1 Model structure

The structure of LTI model proposed in this paper for the movement in a horizontal plane, with constant Z and based on zero Yaw angle assumption, is represented by the following state equations:

$$\dot{x} = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ a_{21} & a_{22} & a_{23} & a_{24} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ a_{41} & a_{42} & a_{43} & a_{44} & 0 & 0 & 0 & 0 \\ a_{51} & a_{52} & 0 & 0 & a_{55} & 0 & 0 & 0 \\ 0 & 0 & a_{63} & a_{64} & 0 & a_{66} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & a_{76} & 0 & 0 \\ 0 & 0 & 0 & 0 & a_{85} & 0 & 0 & 0 \end{bmatrix} x + \begin{bmatrix} 0 & 0 \\ b_{21} & b_{22} \\ 0 & 0 \\ b_{41} & b_{42} \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} \qquad (1)$$

Where:

x_1 : Roll
x_2 : Roll derivative
x_3 : Pitch
x_4 : Pitch derivative

x_5 : Lateral linear speed
x_6 : Longitudinal linear speed
x_7 : Frontal displacement
x_8 : Longitudinal displacement

u_1 : Roll control signal
u_2 : Pitch control signal

This model structure assumes the existence of coupling between the Pitch and Roll angles, and a dependence on the lateral and longitudinal movements only with variations in Pitch and Roll angles respectively. From various experiments, we were able to verify that this structure model does not depend on the value of Z, or different constant values that can take the angle Yaw.

The model structure represented in Equation 1 is a parametric model whose coefficients a_{ij}, b_{kl}, should be defined for a particular aerial vehicle. For this, we developed an identification process on the real system, using genetic algorithms as computational tool.

During early experiments with the mini-helicopter and after data analysis aimed at obtaining a single model to represent its dynamics, we were able to show that its dynamic behavior when flying in a horizontal plane was sufficiently diverse from stationary, frontal and lateral flight. This raised the need to obtain a specific model for each of these types of flight.

2.2.2 Parameters Identification

A genetic-algorithm-based tool, which has been successfully used to parameterize other models (DelCerro 2007), was used for the a_{ij} and b_{kl} parameter's identification process.

The selection criterion used was the minimization of an objective function given by the difference between real signals measured in the VTOL and signals provided by the model. This difference is represented by the value of the mean square error between those signals.

The implementation of the identification process in the case of mini-helicopter resulted in the parameters that are detailed in Table 1, for the three different types of flight.

Table 1. Model parameters for mini-helicopter

Parameter	Hover flight	Longitudinal flight	Lateral flight
a_{21}	-0.0897	-4.9522	-9.4286
a_{22}	-1.0211	-0.1898	-2.8422
a_{23}	1.8807	1.5900	6.1920
a_{24}	0.7150	5.0240	-4.6793
a_{41}	-1.0785	-3.6885	-0.0641
a_{42}	-2.6725	-1.8418	0.4656
a_{43}	-1.1095	-4.4536	-8.1002
a_{44}	-2.0666	-3.4587	-5.5487
a_{51}	0.3604	-0.7884	0.2680
a_{52}	0.0213	0.8237	0.5478
a_{55}	-1.1681	-0.1264	-0.9287
a_{63}	1.2946	-0.4601	-1.3005
a_{64}	3.0741	0.5184	0.3482
a_{66}	-4.4571	-0.9718	-0.1548
a_{76}	-0.2156	-1.8407	-0.4067
a_{85}	0.3313	0.0961	1.1147
b_{21}	1.2969	-5.0342	-12.7259
b_{22}	0.8092	2.8049	-0.0557
b_{41}	-3.6516	-4.4360	1.8356
b_{42}	-1.3202	-4.0981	-3.6725

2.3. Quadrotor model

Similarly to mini-helicopter, was considered the quadrotor as a decoupled system respect to its three main movements: the movement in a horizontal plane parallel to the ground, with varying angles roll, pitch and the resulting lateral and longitudinal displacements, the Yaw movement and vertical movement or Z variation. However, given the geometrical and dynamic characteristics of quadrotor no distinction is made between stationary, lateral and longitudinal flights, and only one LTI model was obtained, which represents the overall dynamic behavior of the vehicle.

2.3.1. Quadrotor model structure

Given the physical limitations of quadrotor's payload and the consequent difficult to obtain reliable measures of variables such as the position (X, Y, Z) a simple model to represent a smaller number of state variables with respect to mini-helicopter was chosen.

We propose a LTI, fourth order model (Equation 2) that will be useful to represent the Pitch and Roll angles, and then develop attitude controllers to provide stability to dynamics of quadrotor.

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \\ \dot{x}_4 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ a_{21} & a_{22} & a_{23} & a_{24} \\ 0 & 0 & 0 & 1 \\ a_{41} & a_{42} & a_{43} & a_{44} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ b_{21} & b_{22} \\ 0 & 0 \\ b_{41} & b_{42} \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} \qquad (2)$$

Where:

x_1 : Roll
x_2 : Roll derivative
x_3 : Pitch
x_4 : Pitch derivative

u_1 : Roll control signal
u_2 : Pitch control signal

2.3.2. Parameters Identification

Using the same computational tools of mini-helicopter case, the parameters of the LTI model for quadrotor were identified, whose values are summarized in Table 2.

Table 2. Parameters model for Quadrotor

Parameter	Value
a_{21}	-1.1132
a_{22}	-0.7118
a_{23}	0.0001
a_{24}	0.0000
a_{41}	7.2721
a_{42}	6.1794
a_{43}	-7.9908
a_{44}	-5.2474
b_{21}	-1.1045
b_{22}	0.0000
b_{41}	-4.5267
b_{42}	-14.4146

3. Models Validation

3.1 Mini-helicopter's model.

Some graphics are shown to compare the results of the LTI model simulations and the real data obtained in different experimental flights. Figures 2, 4 and 6 show, on the left side, the comparison between the real data (solid lines) and those resulting from the model simulations (dotted lines) with the same set of data used in the parameters identification process. In the right side is reported the same comparison between real and simulated data, but with a different set of data called "control data".

Figures 2, 4 and 6 compare real values of Pitch and Roll angles, as well as lateral and longitudinal movements of mini-helicopter (which coincide with the X and Y axes, when Yaw angle is zero), with those resulting from the simulation of stationary, lateral and longitudinal flights, obtained from the respective models previously described.

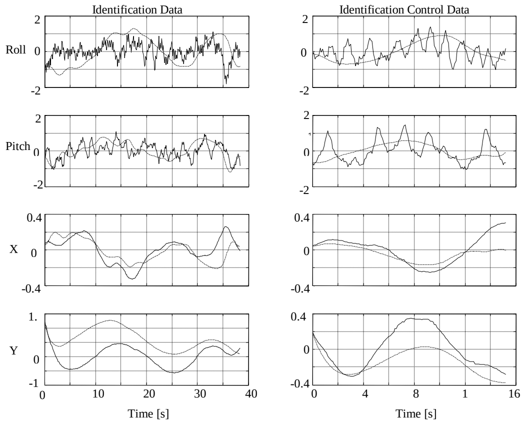


Figure 2: Real and Simulated Data Comparison for Hover Flight.

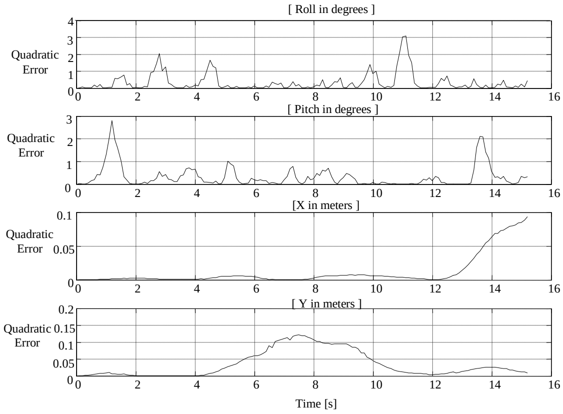


Figure 3: Quadratic Error for Data Control in Hover Flight

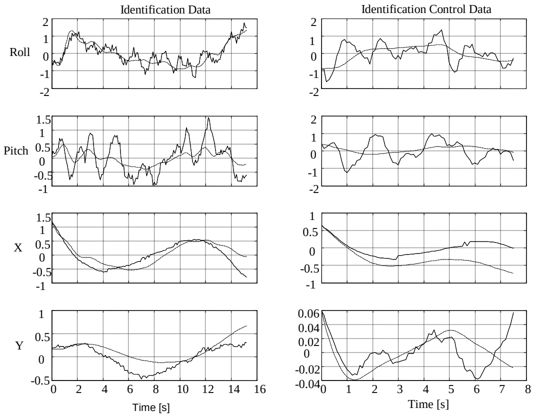


Figure 4: Real and Simulated Data Comparison for Longitudinal Flight

The plots in Figures 2, 4 and 6 shows that real Roll and Pitch signals have a higher frequency component than those obtained from the model simulation. However, frequency analysis of signals measured were performed with the mini-helicopter in flight, with the mini-helicopter on the ground and engine turned on and then with the engine off. Such analysis could determine that these high-frequency signals are due to mechanical vibrations caused by the engine movement.

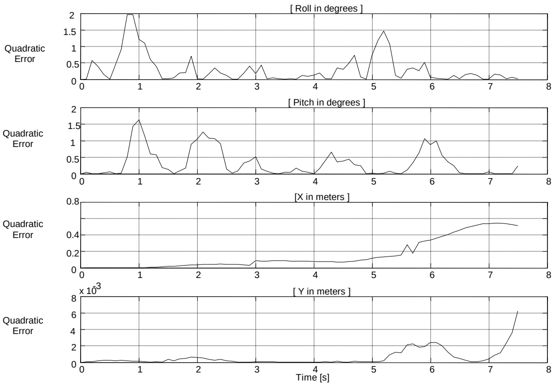


Figure 5: Quadratic Error for Control Data in Longitudinal Flight

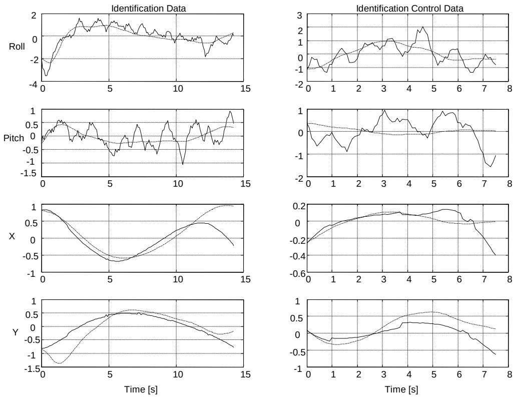


Figure 6: Real and Simulated Data Comparison for Lateral Flight

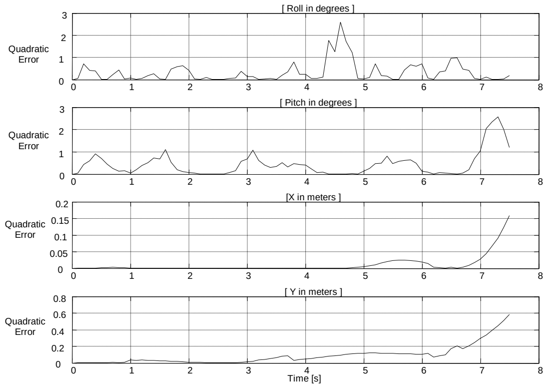


Figure 7: Quadratic Error for Control Data in Lateral Flight

The quadratic error graphics (Figures 3, 5 and 7) show differences smaller than 0.5 degrees between real and simulated data, in Pitch and Roll angles. The resulting behavior on the position in X and Y also presents a negligible error, except for some specific samples where the error Pitch and Roll is a little higher (in all cases of flight). This error may be attributed to high frequency vibrations commented previously.

It should be noted that the latest samples from each data sets are part of a transition phase between the three different types of flight (stationary, front and side), which explains that towards the end of the quadratic error graphics, this error becomes bigger, because the actual behavior of the helicopter in these transitions does not correspond exactly with any type of flight consideration.

With the above can be concluded that model structure, with the corresponding sets of parameters obtained, is a suitable representation of mini-helicopter dynamic behavior. In other words, the model, with relevant parameters depending on type of flight, allows an approximate representation of mini-helicopter complex dynamic. For this reason, this model can serve as a practical and useful tool for driver design and development procedures, despite the limitations and approaches set out in the model definition (LTI model with relatively small order).

3.2 Quadrotor model.

Figure 8 shows the same comparisons for the case of the quadrotor. On the left side real data (solid line) is compared to model simulation (dotted line) on the set of data used in the identification process. On the right side the same comparison using a different set of data (control data).

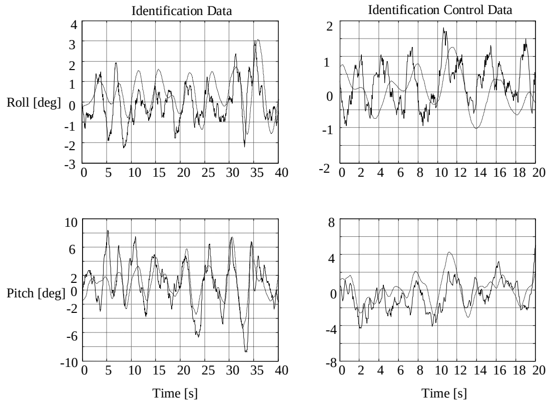


Figure 8: Comparison of Real and Simulated Data for the Quadrotor Model.

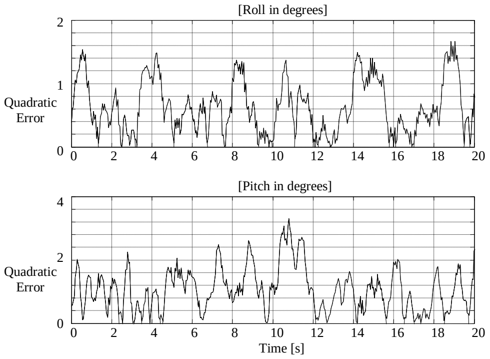


Figure 9: Quadratic Error of the Control Data in the Quadrotor Model

Figure 9 shows the difference between the measure and simulated Roll and Pitch values, expressed as quadratic error. As it can be noticed, the difference is bigger than in the case of the helicopter. This is due to the fact that in the quadrotor case only one model has been used for the three flight modes. Moreover, due to the limited payload of this vehicle, an IMU with less precision had to be used.

4. Design procedure obtained

Once obtained the models for the two vehicles described above, it was possible to establish a standard procedure for the design of analytical controllers with many proven techniques, such as pole placement, linear quadratic regulators and sliding modes control. This is the main advantage of our modeling methodology, which was validated by the design of various controllers for both UVTOLs.

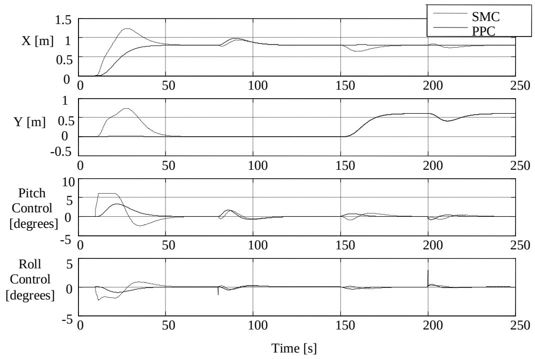


Figure 10: Comparison Between Two Different Control Techniques for the Mini-helicopter.

Figure 10 shows the comparison between X and Y position signals for the mini-helicopter, in response to small changes in the reference for maintaining it in stationary flight, controlled using sliding mode control (SMC) and pole placement control (PPC).

In the same way, Figure 11 shows simulations of quadrotor's Roll and Pitch regulated using pole placement control (PPC), sliding mode control (SMC) and LQR control over the quadrotor.

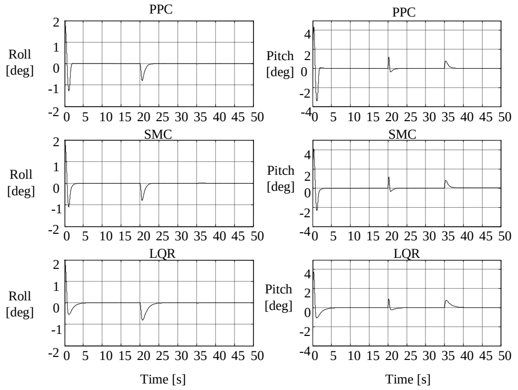


Figure 11: Comparison Between Simulations of Three Different Control Techniques for the Quadrotor.

Finally, in order to compare the goodness of the controller modelling and design process, we have realized some experimental flight with the quadrotor using the pole placement control technique for controlling the Roll and Pitch signals, achieving stationary attitude for the vehicle as shown in Figure 12. The third graph of Figure 12 shows the signal command for height (Z) to show how long the quadrotor was in the air.

Note that the Pitch values do not reach the value zero due to the position of the IMU that involve an offset for this variable. In general, the controller designed for this case allow a good behaviour, compare with manual control, including reject of perturbation that can be noticed at t = 8.5 s.

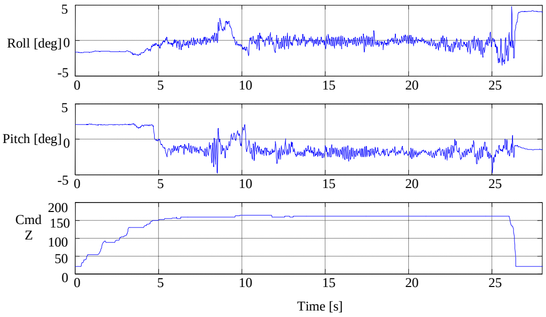


Figure 12: Results Obtained for Sliding Mode Control for the Quadrotor.

5. CONCLUSION

We have developed and tested a simple methodology that allows obtaining LTI models for small VTOL vehicles, as well as to design implement and test different kinds of controllers for them. This methodology can be summarized as follows:

- Perform manual flights with the VTOL to get their input-output data (Roll, Pitch, V_{Roll}, V_{Pitch}, V_X, V_Y, X, Y)
- Feed the measured data to software designed to get the LTI model parameters.
- Validate the model based on simulations, comparing it with actual data.
- Define the design criteria to use and, with the control technique desired, to obtain the controller design.
- Simulate the controlled system and tune the controller parameters, test the performance of UVTOL using Simulink®
- Proceed to rapid prototyping controller using Simulink®, getting the code in C or C++.
- Test the controller in the UVTOL architecture.

ACKNOWLEDGMENTS

The authors would like to thank Ministerio Educación y Ciencia (Spain) by the FRACTAL project (DPI 2006_03444) and Pontificia Universidad Javeriana Cali (Colombia) for founding part of this research work.

REFERENCES

Aguirre, I., 2001. *Identificación y control de un helicóptero en vuelo estacionario*. Thesis (PhD). Universidad Politécnica de Madrid, Spain.

Amir, M.Y., Abbass, V., 2008. Modeling of Quadrotor Helicopter Dynamics. International Conference on Smart Manufacturing Application. April. 9-11. KINTEX, Gyeonggi-do, Korea

Avila, J.C., Brogliato, B., Dzul, A., Lozano, R., 2003. Nonlinear modelling and control of helicopters. *Automatica*, 39. 1583 – 1596.

Bouabdallah, S., Siegwart, R., 2005. Backstepping and Sliding-mode Techniques Applied to an Indoor Micro Quadrotor. *Proceedings of the 2005 IEEE International Conference on Robotics and Automation*. April. Barcelona, Spain.

Castillo, M., Alvis, W., Castillo, C., Valavanis, K., Moreno, W., 2005. Small Scale Helicopter Analysis and Controller Design for Non-Aggressive Flights. *2005 IEEE SMC Conference.*

Castillo, P., Dzul, A., Lozano, R., 2004. Real-Time Stabilization and Tracking of a Four-Rotor Mini Rotorcraft. *IEEE Transactions on Control Systems Technology*, 12 (4).

Cunha, R., Silvestre, C., 2003. SimModHeli: A dynamic simulator for modelscale helicopters. *Proc. MED2003 - 11th Mediterranean Control on Control and Automation*. Rodes, Greece.

DelCerro, J., Valero, J., Barrientos, A., 2004. Modelling and Identification of a Small Unmanned Helicopter. *World Automation Congress (WAC ISORA).* Seville, Spain.

DelCerro, J., 2007. *Arquitectura abierta para el control autónomo y teleoperado de un mini-helicóptero.* Thesis (PhD). Universidad Politécnica de Madrid, Spain.

Isidori, A., Marconi, L., Serrani, A., 2003. Robust Nonlinear Motion Control of a Helicopter. *IEEE Transactions on Automatic Control*. 48 (3).

La Civita, M., Messner, W., Kanade, T., 2002. Modeling of Small-Scale Helicopters with Integrated First-Principles and System-Identification Techniques. *Proceedings of the 58th Forum of the American Helicopter Society*, 2505-2516, June. Montreal, Canada.

Madani, T., Benallegue, A., 2006. Control of a Quadrotor Mini-Helicopter via Full State Backstepping Technique. Proceedings of the 45th IEEE Conference on Decision & Control. December 13-15. San Diego, CA, USA.

McGeoch, D.J., McGookin, E.W., 2005. MIMO Sliding Mode Attitude Command Flight Control System for a Helicopter. *AIAA Guidance, Navigation and Control Conference and Exhibit.* August. San Francisco, California.

Vélez, C.M., Agudelo, A., Álvarez, J., 2006. Modeling, Simulation and Rapid Prototyping of an Unmanned Mini-Helicopter. *AIAA Modeling and Simulation Technologies Conference and Exhibit*. 21 - 24 August. Keystone, Colorado.

Xu, R., Ozguner, U., 2006. Sliding Mode Control of a Quadrotor Helicopter. *Proceedings of the 45th IEEE Conference on Decision & Control.* December 13-15. San Diego, CA, USA.

ON THE INTEGRATED PRODUCTION AND PREVENTIVE MAINTENANCE PROBLEM IN MANUFACTURING SYSTEMS WITH BACKORDER

Kenne, J.P[(a)], Gharbi, A[(b)], Beit, M[(c)]

[(a)] Mechanical Engineering Department, Laboratory of Integrated Production Technologies
Université du Québec, École de technologie supérieure,
1100, Notre Dame Street West, Montreal (Quebec), Canada, H3C 1K3

[(b)] Automated Production Engineering Department, *Production Systems Design and Control Laboratory*
Université du Québec, École de technologie supérieure,
1100, Notre Dame Street West, Montreal (Quebec), Canada, H3C 1K3

[(a)]jean-pierre.kenne@etsmtl.ca, [(b)]ali.gharbi@etsmtl.ca, [(c)]mouhir.beit.1@ens.etsmtl.ca

ABSTRACT

The integrated production, inventory and preventive maintenance problem (PIPMP) is concerned with coordinating production, inventory and preventive maintenance operations in order to meet customer demand with the aim of minimizing costs. A unified framework is developed allowing production and preventive maintenance to be jointly considered using an age-dependent optimization model, itself based on the minimization of an overall cost function; this cost function for its part includes inventory holding, backlog, and preventive and corrective maintenance costs. We provide optimality conditions for more realistic manufacturing systems and use numerical methods to obtain the optimal preventive maintenance policy and the relevant age-dependent or multiple-threshold-levels production policy, which we refer to as the *multiple threshold levels hedging point policy*. Numerical examples are included to illustrate the importance and the effectiveness of the proposed methodology

Keywords: Preventive maintenance, Buffer inventory, Backorder, Reliability theory, Manufacturing Systems.

1. INTRODUCTION

Preventive maintenance involves a schedule of planned maintenance actions aimed at the prevention of breakdowns and failures. The long-term benefits of preventive maintenance include: improved system reliability, decreased replacement cost, decreased system downtime and better spares inventory management. The aim of this paper is: (i) to propose a probabilistic control model for the simultaneous planning of the production and preventive maintenance of a manufacturing system, and (ii) to develop an efficient technique for the computation of the optimal control policy considered.

An overview of relevant literature reveals that significant contributions have been proposed based on (i) preventive maintenance, (ii) production control, and (iii) joint production and maintenance optimization models. It has been shown in [10] and [11] that policies which do not address maintenance and production control decisions in an integrated manner can perform quite poorly. Available models are considered individually or simultaneously, and are restricted to simplified assumptions that sometimes lead to not-so-realistic preventive maintenance or production policies. Comparisons of long-term effects and costs usually favour preventive maintenance over the performance of maintenance actions only when the system fails. The age replacement policy (ARP) is one of the available preventive maintenance options, and takes precedence over the block replacement policy (BRP) or group replacement policy (GRP). For details on these policies, please see [3] and [1] and their corresponding references. Details on other maintenance policies and their effects on the productivity and availability of a manufacturing system can be found in [17]. With the age replacement policy, which is a basic and simple replacement policy, the unit is replaced upon failure or at a preset age, whichever occurs first [9]. A generalized age-replacement policy with age-dependent minimal repair and random lead-time is presented in [18] by considering the average cost per unit time and the stochastic behavior of the system considered. The model includes the cost of storing a spare as well as the cost of system downtime. However, the implementation of an age replacement policy requires the continuous tracking of a component's service life. This explains the popularity of the block replacement policy (BRP) in industries with large systems, each having a specific number of components. Given that ARP is based on age-dependent preventive maintenance periods rather than fixed periods, as is the case with BRP, it remains more realistic, and thus attracts many researchers. We refer the reader to extended versions of age replacement policies and their implementation presented in [1]. The related policies are non-realistic in the context of manufacturing systems, given that frequent machine breakdowns inevitably create bottlenecks for the process. Hence, preventive maintenance (used to reduce the likelihood of machine breakdowns) combined with the control of finished goods inventories, is a potential means of reducing the overall incurred cost.

The dynamics of the finished goods inventory are not considered in the aforementioned models, which are classified here as static models, given that the policies obtained are based on the mean values of the stochastic processes involved. Manufacturing systems with unreliable machines have been modeled using the stochastic optimal control theory, in which failures and repairs processes are supposed to be described by homogeneous Markovian processes. The related optimal control model falls under the category of problems presented in the pioneering work of Rishel [15] and in [4]. The analytical solution of the one-machine, one-product manufacturing system presented in [2] result from investigations carried out in the same direction. Preventive maintenance planning problems are combined with production control to increase the availability of the production system, and hence to reduce the overall incurred cost [5].

The preventive maintenance model for a production inventory system is developed in [7] using information on system conditions (such as finished product demand, inventory position, costs of repair and maintenance, etc.) and a continuous probability distribution characterizing the machine failure process. An analytical model of the BRP and safety stock strategy is formulated in [14], also using restrictive assumptions such as the fact that the time to accomplish build-up and depletion of safety stock is small relative to the mean time to failures (MTTF). The model presented in [16] combines the ARP with the safety stock to show that inventory needs to be built just before the preventive maintenance occurs. It is assumed in [16] that extra capacity is maintained in order to hedge against the uncertainties of the production processes, and that there will be no possible breakdown of the machine before the preventive maintenance date. Without the assumption made in [16] on machine dynamics, the stochastic optimal control theory is used in [6], [12] and [13] to define machine age-dependent production and preventive maintenance policies. Such policies are based on increasing failure rate (IFR) distributions, and are characterized by a staircase structure, but with only one step.

The purpose of this paper is to investigate the joint implementation of preventive maintenance and safety stocks in a manufacturing environment in the presence of more realistic features than those made in [14] and [16]. The model presented in this paper is applied to a manufacturing system capable of catching up with unmet demand without interrupting the normal production process, as soon as production resumes (backorder situation). We consider the possibility of having a breakdown during the catch-up period. Previous models in the literature assumed that such a possibility is negligible. In addition, there is no restriction on any of the operational, repair and preventive maintenance time distributions (i.e., there is no restriction regarding the exact type of distribution of the time to machine breakdown, the time to corrective maintenance and the time to preventive maintenance).

The optimal production and preventive maintenance policies obtained in this paper significantly reduce the incurred cost, and are shown to be characterized by a multiple-step staircase structure describing the fact that the stock threshold level increases with the machine age given that its failure probability also increases with age. Significant stock levels at high machine ages are used to hedge against more frequent failures that occur randomly in such situations. The staircase structure of the control policy is shown to be the major contribution of the proposed model. The performance of such a policy is compared to that of the single stock threshold level control policy presented in [2] and extended in [12]. The model presented here is developed through the characterization of the stochastic processes underlying the system. Using the properties of the probability structure of these stochastic processes, the overall incurred cost is considered as a performance criterion, and is used as a basis for optimally determining the set of age-dependent stock threshold levels for each operational time or for each range of machine ages.

2. MODEL NOTATIONS AND ASSUMPTIONS

Throughout the article, we will be using the following notations and assumptions:

Notations

ARP	Age Replacement Policy
GFR	General Failure Rate
HPP	Hedging Point Policy
IFR	Increasing Failure Rate
DFR	Decreasing Failure Rate
MADP	Multiple Age-Dependent Policy
MTTF	Mean Time to Failures
MTTR	Mean Time to Repair
SADP	Single Age-Dependent Policy
SIFRR	Single Increasing Failure Rate Region
c^+	Inventory holding cost per unit time
c^-	Penalty cost for each unit of unmet demand
c_1	Cost of corrective maintenance
c_2	Cost of preventive maintenance
$u(\cdot)$	Production rate of the system
$u_{\max}$	Maximal production rate
d	Demand rate
$\zeta(t)$	State of the machine at time t
T_b	The time to machine breakdown
$f(t)$	Probability density function of T_b
$F(t)$	Cumulative distribution function of T_b
$R(t)$	Reliability function of the machine
μ	Mean time to machine breakdown
T_{pm}	Time to preventive maintenance
$q(t)$	Probability density function of T_{pm}
T_{cm}	Corrective maintenance time
$g(t)$	Probability density function of T_{cm}
S	Single stage stock threshold level

S_i Stage *j* stock threshold level or stock level

dA_j interval j of the machine age partition

T Scheduled time to preventive maintenance

Assumptions

The fundamental assumption of ARP, that the cost of failure replacement (c_1) is greater than the cost of preventive replacement (c_2), is also applicable here. Other assumptions considered in this paper are:

1. All failures are instantly detected and repaired.
2. If a machine failure occurs during a production phase, corrective repair is started immediately and after repair, the machine is restored back to the same initial working condition. In addition, a preventive maintenance action (as a corrective one) renews the production system (i.e., the age of the machine is set to zero).
3. The mean value of the time requirement for a preventive maintenance operation is short when compared with the mean time to machine breakdown (i.e., $E[T_{pm}] < E[T_b] = \mu$).
4. A sufficient capacity is present to allow the accumulation of safety stock at the beginning of each machine life cycle.
5. The time to accomplish the build-up and depletion of safety stocks is not necessarily small relative to the MTTF (i.e., a breakdown could arise during that time) and the system is feasible (i.e., $\frac{MTBF}{MTBF + MTTR} \cdot u_{\max} \geq d$)
6. All unmet demand is not lost, as is the case in many others works in the relevant literature.

3. PROBLEM STATEMENT

The manufacturing system considered consists of a single machine which is subject to random breakdowns and repairs. The machine in question can produce one part type and its state can be classified as operational, denoted by 1, under repair, denoted by 2, and under preventive maintenance, denoted by 3. Assume that $\zeta(t)$ denotes the state of the machine with value $M = \{1, 2, 3\}$.

The dynamics of the machine could be described by a stochastic process, with jumps from modes α to mode β, illustrated in Figure 1 by Jump($\alpha \to \beta$). Figure 1, called a transition diagram, describes the dynamics of the machine considered.

The transition from mode (operational mode) to mode 2 (repair mode) is machine age-dependent, and is described by an increasing failure rate (IFR) distribution. The transition from mode 1 to mode 3 (from operational to preventive maintenance modes) is controlled in order to increase the capacity of the considered system characterized by an IFR distribution and an overall cost of maintenance which is lower than the overall cost of a corrective maintenance.

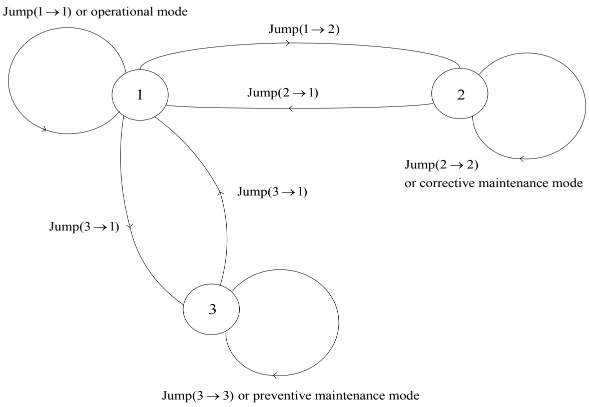

Jump(1 → 1) or operational mode
Jump(1 → 2)
1
Jump(2 → 1)
2
Jump(2 → 2)
or corrective maintenance mode
Jump(3 → 1)
Jump(3 → 1)
3
Jump(3 → 3) or preventive maintenance mode

Figure 1: States transition diagram of system

Based on the costs ratios and inventory level, the proposed model will determine an optimum time for a preventive maintenance action. In a more realistic manufacturing context (no restriction on any of the operational, repair and preventive maintenance time distributions, as mentioned in the Assumptions section), three different distributions are considered in this paper, rather than using a Markov process, as in [8] and references therein.

The system behavior is described by a hybrid state comprising both a discrete and a continuous component. The discrete component consists of the discrete event stochastic processes describing modes 1, 2, and 3. Let us assume that $u(\cdot)$ denotes the production rate of the machine at time t for a given stock level x and time for preventive maintenance T. The feasible production policies set is given by

$$\Gamma = \left\{ \left(u(\cdot) \right) \in \Re,\ 0 \leq u(\cdot) \leq u_{\max} \right\} \tag{1}$$

where $u(\cdot)$ is known as the control variable, and constitutes the so-called control policy of the problem under study. The continuous component consists of a continuous variable $x(\cdot)$ corresponding to the inventory/backlog of products. This state variable is described by the following differential equation:

$$\frac{dx(\cdot)}{dt} = u(\cdot) - d \qquad x(0) = x \tag{2}$$

where *x,* and *d* are the given initial stock level and demand rate respectively. Let $g(\cdot)$ be the cost rate defined as follows:

$$g\left(\alpha(t), x, \cdot\right) = c^+ x^+ + c^- x^- + c_1 \mathrm{Ind}\{\alpha(t) = 2\} + c_2 \mathrm{Ind}\{\alpha(t) = 3\} \tag{3}$$

where c^+ and c^- are costs incurred per unit of produced parts for inventory and backlog respectively, $x^+ = \max(0, x)$, $x^- = \max(-x, 0)$ and

$$\mathrm{Ind}\{\Theta(\cdot)\} = \begin{cases} 1 & \text{if } \Theta(\cdot) \text{ is true} \\ 0 & \text{otherwise} \end{cases} \tag{4}$$

for a given proposition $\Theta(\cdot)$. The corrective and preventive maintenance activities involve constant costs, namely c_1 and c_2 respectively.

Our objective is to control the production rate $u(\cdot)$ so as to minimize the overall total cost, integrating the instantaneous cost given by equation (3).

Whenever a breakdown occurs, corrective maintenance is performed, during a random amount of time, in order to restore the machine to its initial condition (i.e., the machine is assumed to be new, and its age is reset to zero). During the maintenance period, one of the following two situations occurs:

- demands for items are met only through safety sock;
- unmet demands for items are backlogged.

In order to reduce the likelihood of machine breakdown, preventive maintenance is scheduled and combined with production planning such that each time, immediately after the maintenance operation is performed, the machine is restored to its initial working condition.

The machine state moves from modes 1, 2 and 3 according to random variables T_b, T_{cm} and T_{pm} defined as time to machine breakdown, corrective maintenance time and preventive maintenance time. At mode 1 and for given age a and scheduled preventive maintenance period T, the production rate is given by an extended version of the so-called hedging point policy (HPP), defined by a threshold level, which is valid just for the value of the age, called here $S(t)$. Such a policy is given by:

$$u(x) = \begin{cases} u_{\max} & \text{if } x < S(t) \\ d & \text{if } x = S(t) \\ 0 & \text{otherwise} \end{cases} \tag{5}$$

An example of a surplus trajectory is illustrated in Figure 2, for which we assumed a constant threshold $S_{dt}(t)$ for an infinitesimal index dt at age t (i.e., $\lim_{dt \to 0} S_{dt}(t) = S$). The holding cost in such a situation reflects the average inventory held over the period t to $t + dt$. For a given initial inventory and a given final inventory, the average of the two is taken to be the average inventory. In the cases of manufacturing systems considered here, demand is time-homogeneous, at least in expectation, and so this assumption is valid. The demand rate of the product is a known constant whereas the production rate (which is greater than the demand rate) depends on the decision variables $S(t)$ and T. For the seek of simplicity, we use S instead of $S(t)$ in the rest of the paper without loosing the fact that S is age dependent. During production, the machine and surplus dynamic both involve the four scenarios presented in the next section.

4. OPTIMALITY CONDITIONS

Scenario No. 1. There is a breakdown before the scheduled preventive maintenance time, and the repair process involved ends with inventory or at a positive surplus level. The finished goods inventory in such a situation is illustrated in Figure 2. The holding cost in the period [0.T] is computed using area A1-1 illustrated in Figure 2.

$$\text{A1-1} = S \cdot \int_0^T R(t)dt$$

where $R(t)$ is the reliability function of the machine and t_f, as represented in figure 2, is its failure age with $t_f < T$. The holding cost, for scenario 1, is thus given by:

$$\begin{aligned} Cost_{1-1} &= c^+ \cdot \text{A1-1} \\ &= c^+ \cdot S \cdot \int_0^T R(t)dt \end{aligned} \tag{6}$$

The repair process ends at time t_r before the zero inventory point. The subsequent production cycle is undertaken and restarted at zero inventory, as illustrated in Figure 2. This is obtained by letting $s(a) := 0$ in the control policy given by equation (5) until t_0 where $x(t) = 0$. From the failure time t_f to the beginning of the subsequent production cycle t_0, the holding cost involved is computed using area A1-2 illustrated in Figure 2.

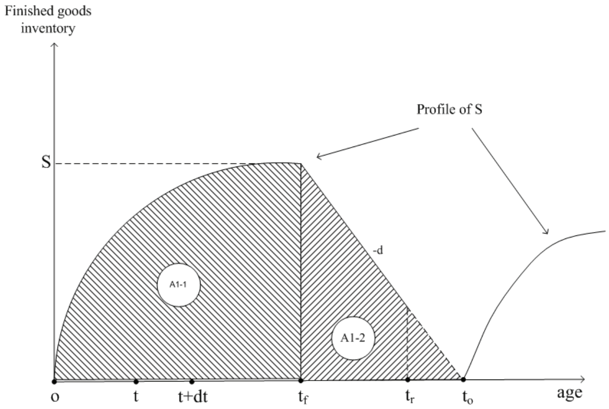


Figure 2: Inventory sample path in the case of failure before preventive maintenance without backlog

The area A1-2 is given by the following expression:

$$\text{A1-2} = \int_0^T \left[\frac{S^2}{2d} \right] dF(t)$$

where $dF(t)$ is the probability of a breakdown occurrence in interval $[t, t + dt]$. The holding cost, for scenario 2, from t_f to t_0, is thus given by:

$$\begin{aligned} Cost_{1-2} &= c^+ \cdot \text{A1-2} \\ &= c^+ \cdot \int_0^T \left[\frac{S^2}{2d} \right] dF(t) \end{aligned} \tag{7}$$

The cost incurred in periods $[0, t_f]$ and $[t_f, t_0]$, for scenario 1, is determined using equations (6) and (7). Such a cost is given by:

$$\begin{aligned} Cost_1 &= Cost_{1-1} + Cost_{1-2} + c_1 \cdot F(T) \\ &= c^+ \cdot S \cdot \int_0^T R(t)dt + c^+ \cdot \int_t^{t+dt} \left[\frac{S^2}{2d} \right] dF(t) + c_1 \cdot F(T) \end{aligned} \tag{8}$$

Scenario No. 2. There is a breakdown before the scheduled preventive maintenance time, and the repair process ends with a backlog situation or at a negative surplus level. The finished goods inventory in such a situation is illustrated in Figure 3. The holding cost in scenario 2 is given by equation (8), and the backlog cost is described by proposition 1.

Proposition 1. By multiplying the number of backlogged unmet demand units by a penalty cost, we obtain the following backlog cost for scenario 2.

$$Cost_2 = c^- \cdot A2 = c^- \cdot \int_0^T \left[\left(\int_{\frac{s}{d}}^{\infty} \left(x - \frac{S}{d} \right) \cdot g(x)dx \right)^2 \cdot \frac{d}{2} \cdot \left(\frac{u_{\max}}{u_{\max} - d} \right) + c_1 \right] dF(t) \tag{9}$$

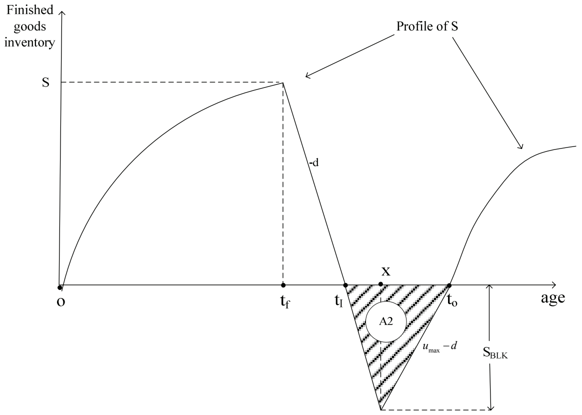


Figure 3: Inventory sample path in the case of failure before preventive maintenance with backlog

Proof: From Figure 3, we can write the following expressions:

$$t_l - t_f = \frac{S}{d} \tag{10}$$

$$x - t_l = \int_{\frac{s}{d}}^{\infty} \left(x - \frac{S(t)}{d} \right) \cdot g(x)dx \tag{11}$$

$$\begin{aligned} S_{BLK} &= d \cdot (x - t_l) \\ &= (t_0 - x) \cdot (u_{\max} - d) \end{aligned} \tag{12}$$

Using equations (10), (11) and (12), the time needed to produce parts for backlogged demand units (i.e., from x to t_0) is given by:

$$t_0 - x = \frac{d}{u_{\max} - d} \cdot \int_{\frac{s}{d}}^{\infty} \left(x - \frac{S(t)}{d} \right) \cdot g(x)dx \tag{13}$$

From Figure 3 (see area $A2$), $t_0 - t_l = (t_0 - x) + (x - t_l)$. Equations (11) and (13) give the following expression:

$$t_0 - t_l = \frac{u_{\max}}{u_{\max} - d} \cdot \int_{\frac{s}{d}}^{\infty} \left(x - \frac{S(t)}{d} \right) \cdot g(x)dx \tag{14}$$

Area $A2$ is then given by:

$$\begin{aligned} A2 &= \frac{S_{BLK} \cdot (t_0 - t_l)}{2} \\ &= \frac{d \cdot u_{\max}}{2(u_{\max} - d)} \cdot \left(\int_{\frac{s}{d}}^{\infty} \left(x - \frac{S(t)}{d} \right) \cdot g(x)dx \right)^2 \end{aligned} \tag{15}$$

Finally, the backlog cost in scenario 2 is

$$Cost_2 = c^- \cdot \int_0^T A2dF(t) + c_1 F(T) \tag{16}$$

For scenarios 1 and 2, a corrective maintenance cost is added at the end of the cycle, given that the machine fails before the scheduled preventive maintenance.

Scenario No. 3. There is no breakdown before the scheduled preventive maintenance time, and the maintenance process ends with an inventory or positive surplus level.

A preventive maintenance is performed on the machine at the scheduled time T (Figure 4), and is completed at $T + T_{pm}$ as illustrated in Figure 4.

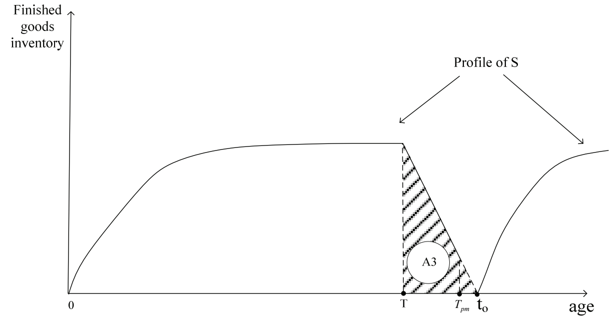


Figure 4: Inventory sample path in the case of no failure before preventive maintenance without backlog

For this scenario, there is no failure before T and the holding cost is computed as in scenario 1, evaluated at T (i.e., for $t = T$), and added to the preventive maintenance cost. The cost is then given by:

$$Cost_3 = c^+ \cdot A3 = c^+ \cdot S \cdot \left(\frac{S^2}{2d} + \int_0^T R(t)dt \right) + c_2 \cdot R(T) \tag{16}$$

Scenario No. 4. There is no breakdown before the scheduled preventive maintenance time, and the maintenance process ends with a backlog or negative surplus level.

A preventive maintenance is performed on the machine at the scheduled time T and is completed at $x = T + T_{pm}$ as illustrated in Figure 5. The holding cost is computed as previously, evaluated at T (i.e., for $t = T$). The backlog cost for scenario 4 is:

$$Cost_4 = c^- \cdot A4 = c^- \cdot \frac{d \cdot u_{\max}}{2(u_{\max} - d)} \cdot \left(\int_{\frac{s}{d}}^{\infty} \left(x - \frac{S}{d} \right) \cdot q(x)dx \right)^2 + c_2 \cdot R(T) \tag{17}$$

We may recall that the machine age is reset to zero after each operation on the machine (corrective or preventive maintenance).

Using expressions developed through scenarios 1 to 4, the overall cost $L(S,T)$, for one production cycle, ended after a maintenance action, depending on variables S and T, is given by the following expression:

$$\begin{aligned} C(S,T) = & \int_0^T \left[c^+ \cdot S \cdot R(t) \cdot dt \right. \\ & + \int_0^T \left[c^+ \cdot \frac{S^2}{2d} + c^- \cdot \left(\int_{\frac{s}{d}}^{\infty} \left(x - \frac{S}{d} \right) \cdot g(x) dx \right)^2 \cdot \frac{d}{2} \cdot \left(\frac{u_{\max}}{(u_{\max} - d)} \right) \right] \Bigg] \cdot dF(t) + c_1 F(T) \\ & + c^- \cdot \frac{d \cdot u_{\max}}{2(u_{\max} - d)} \cdot \left(\int_{\frac{s}{d}}^{\infty} \left(x - \frac{S}{d} \right) \cdot q(x) dx \right)^2 + c_2 R(T) \end{aligned} \tag{18}$$

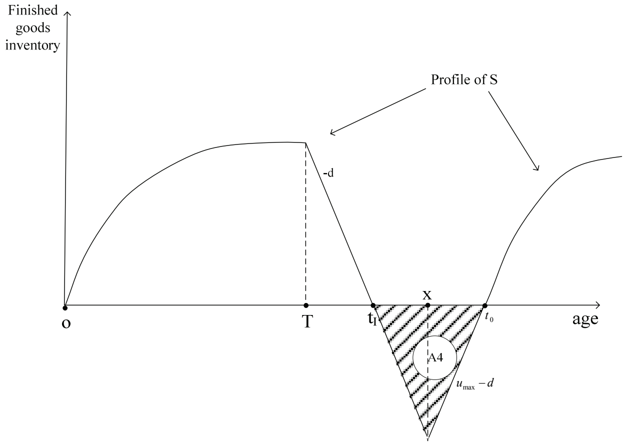


Figure 5: Inventory sample path in the case of no failure before preventive maintenance with backlog

According to According to assumption 2, the duration of a production cycle is approximated given by

$$Cycle\_Time = m(T) + R(T) \cdot MTTP + F(T) \cdot MTTR \tag{19}$$

where The mean operational time of the machine is described by $m(T)$ given by:

$$m(T) = \int_0^T t \cdot f(t) \cdot dt + R(T) \cdot T \tag{20}$$

By dividing $C(S,T)$ by the production cycle time, one obtains the overall expected cost per unit time $L(\cdot)$, given by the following equation:

$$L(S,T) = \frac{C(S,T)}{m(T) + R(T) \cdot MTTP + F(T) \cdot MTTR} \tag{21}$$

In previous scenarios, in order to develop the expression of the overall cost given by equation (16), we assumed the following behavior for the stock:

- The surplus is null at the beginning of the production period, just after an operation on the machine (corrective or preventive maintenance) or at the beginning of the production horizon;
- After an operation on the machine, if the surplus is positive, the production process needs to be stopped and reset to zero inventory using the production policy given by equation (5), and updated here using the aforementioned considerations.

Let us now develop the mathematical model and the optimality conditions to determine values of optimal stock level and maintenance period $S^*(a)$ and $T^*(a)$ for machine age a.

Proposition 2. The function $L(S,T)$ given by equation (16) is convex in S and T.

Proof. Since the holding and backlog cost function is convex and the convexity is maintained under expectation (cf. integration), the proposition follows.

Proposition 2 and previous scenarios are used to show that the optimality conditions, and hence, the age-dependent optimal values of the control policy (threshold levels and scheduled preventive maintenance times), are given by Theorem 1.

Theorem 1. A safety stock S^* and a scheduled preventive maintenance time T^* are optimal at time t if and only if:

$$\frac{\partial L(S,T)}{\partial T} = 0 \quad \text{and} \quad \frac{\partial L(S,T)}{\partial S} = 0$$

For the safety stock, we have:

$$\begin{aligned} \frac{\partial L(\cdot)}{\partial S} = c^+ \cdot \int_t^{t+\Delta t} R(t) \cdot dt + \Delta F(t) \cdot & \left[c^+ \cdot \frac{S}{d} + c^- \cdot \left[-\frac{S}{d} \cdot g\left(\frac{S}{d}\right) - \frac{q\left(\frac{S}{d}\right)}{d} + \frac{S}{d} \cdot g\left(\frac{S}{d}\right) \right] \times \right. \\ & \left. \left[\left(\int_{\frac{s}{d}}^{\infty} \left(x - \frac{s}{d} \right) \cdot g(x) \cdot dx \right) \cdot \frac{d}{2} \cdot \left(1 + \frac{d}{(U_{mx} - d)} \right) \right] \right] = 0 \end{aligned} \tag{22}$$

and certainly,

$$S - 2 \cdot \frac{c^-}{c^+} \cdot F\left(\frac{s}{d}\right) \cdot \left[\int_{\frac{s}{d}}^{\infty} \left(x - \frac{s}{d} \right) \cdot g(x) \cdot dx \right] \cdot \frac{d}{2} \cdot \left(1 + \frac{d}{(U_{mx} - d)} \right) = -\frac{d \cdot \int_t^{t+\Delta t} R(t) \cdot dt}{\Delta F(t)} \tag{23}$$

Proof. In the interval $[0,T]$, the first-order sufficient condition to obtain the threshold value S is $\frac{\partial L(\cdot)}{\partial S} = 0$ with function $L(\cdot)$ described by equation (21). Equation (22) is obtained, and the threshold value or optimal safety stock in $[0,T]$ is given by equation (23). Due to the complexity of the expression (21) with respect to T, obtaining the analytical expression for such a parameter, as in the case of the parameter S, becomes more complex. Hence, we use the convexity property of the function $L(\cdot)$ to complete the proof of this theorem for the parameter T.

We solve equation (23) for S and use a numerical search over a given computational grid to provide the minimal cost and the related control parameters (S for production policy and T for preventive maintenance policy).

5. NUMERICAL PROCEDURE

The time to system failure T_b has a Weibull (β,η) distribution. Here, β and η are the shape and scale (characteristic life) parameters of the distribution. We must recall that a Weibull distribution is IFR when $\eta \geq 1$, and decreasing failure rate (DFR) when $0 < \eta \leq 1$; when $\eta = 1$, $h(t) = 1/\eta = \lambda$, we obtain the exponential distribution which is both IFR and DFR. In this paper, we consider a *Rayleigh* distribution, which is a special case of Weibull, with $\eta = 2$ (i.e., IFR with $h(\infty) = \infty$).

The time to corrective maintenance T_{cm} and the time to preventive maintenance T_{pm} have a Lognormal (μ,σ) distribution, where the two parameters, μ and σ, are the mean and standard deviation of the natural logarithm of the time to perform the operation, corrective or preventive maintenance, represented here by Lognormal random variables. The Lognormal distribution is a single increasing failure rate region (SIFRR), which is a special class of general failure rate (GFR), given that it only has one IFR region ($h(\infty) = 0$).

Given the previous probability distribution functions, the values of $L(S,T)$ were derived from numerical integration methods (Theorem 1) and from the proposed algorithm, to obtain optimal values of $L(\cdot)$, S and T (denoted as $L^*(\cdot)$, S^* and T^* henceforth). An algorithm for the optimal production and preventive maintenance planning problem is given in Figure 6, and was coded using the Matlab/Simulink software.

Based on the proposed algorithm, the numerical scheme proceeds as follows: (i) read input data; (ii) consider computational grid on T for given lower and upper bounds $T^{\min}$ and $T^{\max}$ respectively; (iii) for each feasible scheduled preventive maintenance time T (i.e., $T^{\min} \leq T \leq T^{\max}$), consider a discrete time interval dt and solve the optimality condition at time t to obtain the optimal cost and the associated threshold level.

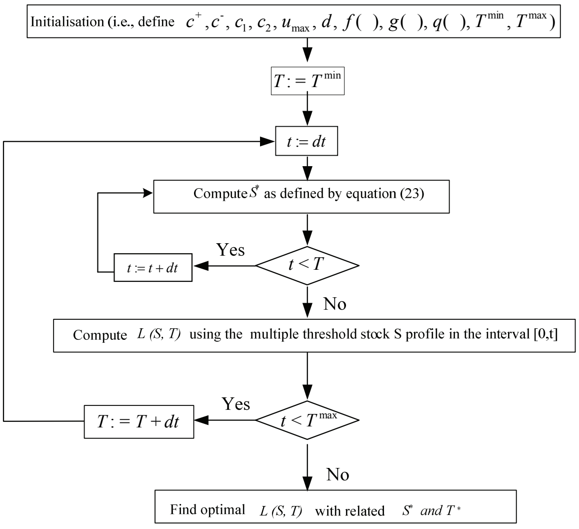

Initialisation (i.e., define $c^+, c^-, c_1, c_2, u_{\max}, d, f(\ ), g(\ ), q(\ ), T^{\min}, T^{\max}$)
$T := T^{\min}$
$t := dt$
Compute S^* as defined by equation (23)
Yes
$t := t + dt$
$t < T$
No
Compute $L\ (S, T)$ using the multiple threshold stock S profile in the interval [0,t]
Yes
$T := T + dt$
$t < T^{\max}$
No
Find optimal $L\ (S, T)$ with related S^* and T^*

Figure 6: Iterative numerical procedure or algorithm

For illustration purposes, assume $u_{\max} = 1$ item per unit of time and the production process is run to satisfy a constant demand rate $d = 0.5$ item per unit of time. The time to breakdown T_b is Weibull, with $\beta = 2$ and $\eta = 100$ (i.e., $\mu = 88.6$). The time to repair and the time to preventive maintenance are Lognormal, with $\mu_{cm} = 20$, $\sigma_{cm} = 1$ and $\mu_{pm} = 10$, $\sigma_{pm} = 0.5$ respectively. For such a system, a sensitivity analysis is provided in the next section to show the usefulness of the proposed approach and to illustrate the contribution of this paper.

6. RESULTS ANALYSIS

Four classes of studies are considered by using the variations of inventory, backlog, corrective and preventive maintenance costs. For the first class, Figure 7 represents the effect of the holding cost variation on the stock level and the preventive maintenance time. For classes 2, 3 and 4, we illustrate the sensitivity analysis through Figures 8 to 10.

It is interesting to note the following from Figure 7, obtained with the variation of the inventory cost c^+ ($c^+ = 2,5,7$ and $c^+ = 1,\cdots,45$ as in figure 7(a) and 7(b) respectively) and $c_1 = 5000,\ c_2 = 3000,\ c^- = 50$:

- The length of the first age interval increases with an increase in the inventory cost (i.e., $dA_1 = 0$ for $c^+ = 2$, $dA_1 = 18$ for $c^+ = 5$ and $dA_1 = 24$ for $c^+ = 7$). This reduces the surplus at small age values, and hence reduces the overall incurred cost (see Figure 7(a)).
- The scheduled production time for preventive maintenance T^* decreases with the increase in the inventory cost due to the fact that the safety stock decreases, and hence increases the

possibility of having a backlog situation (see Figure 7(b)).

It is clear from Figure 7 that the inventory policy significantly influences the overall incurred cost and that the safety stock increases with the machine age for each value of c^+ but with a different trend. The preventive maintenance policy is also significantly affected by the holding cost according to a staircase trend.

For the variation of the backlog cost, Figure 8(a) shows that the stock threshold levels increase with the backlog costs for given machine age intervals. In addition, the length of the first age interval (i.e., dA_1) decreases with the increase in the backlog cost. This increases the surplus at small age values, and hence increases the overall incurred cost, but the occurrence of backlog situations is minimized. It can be noted from Figure 8(b) that the preventive maintenance frequency also increases with the increase in backlog costs, thus increasing the availability of the production process, and hence the avoidance of backlog situations.

The variation of the corrective maintenance affects the preventive maintenance policy, and hence the production policy, as illustrated in Figure 9(a). The preventive maintenance frequency increases with the increase in corrective maintenance costs, as illustrated in Figure 9(b). This is done in order to reduce the number of machine breakdowns, and hence avoid corrective maintenance, which involve excessive costs. The variation of the preventive maintenance cost affects the preventive maintenance policy asymptotically (i.e., constant preventive maintenance for large costs) and hence, the production policy (operation duration) as shown in Figures 10(b) and 10(a), respectively. The results obtained show that such a variation has no significant effect on the staircase trend of the production policy (see Figure 10(a)), as in the case of corrective maintenance cost variations compared to the staircase trends illustrated in Figures 7(a) and 7(a). The preventive maintenance frequency also decreases with an increase in preventive maintenance costs, avoiding frequent preventive maintenance action, and hence reducing the total incurred cost.

7. OPTIMAL CONTROL POLICY

The results obtained, based on previous sensitivity analyses, are satisfactory and practical. In this paper (i.e., backlog case), we compare the results obtained, based on the multiple stage age-dependent control policy, to those presented in. [12], which are based on a single stock age-dependent threshold value. Table 1 shows that the proposed control policy performs well compared to a single threshold-based control policy. The cost reduction given by the proposed control policy ranges between 17% and 37% for different situations generated through a sensitivity analysis. For illustrative purposes, we include a few cases, rather than presenting all cases considered.

To show that there exists an asymptotic trend of variation of stock threshold values, we consider a situation without preventive maintenance (i.e., large value of scheduled preventive maintenance time or $T \to \infty$) and obtain the result presented in Figure 11.

A constant and large threshold level needs to be maintained when the machine age is advanced (i.e., the machine is supposed to be old), in which case the down probability is likely to be 1. The age-dependent production policy obtained, called here the MADP, is an extension of the modified hedging point policy presented in [6] and [12]. The production rate associated with the proposed MADP is given by:

$$u_j(x, age) = \begin{cases} u_{max} & \text{si } x < S_j \text{ and } age \in dA_j \\ d & \text{si } x = S_j \text{ and } age \in dA_j , \quad j = 1, \ldots, k \\ 0 & \text{si } x > S_j \text{ and } age \in dA_j \end{cases}$$

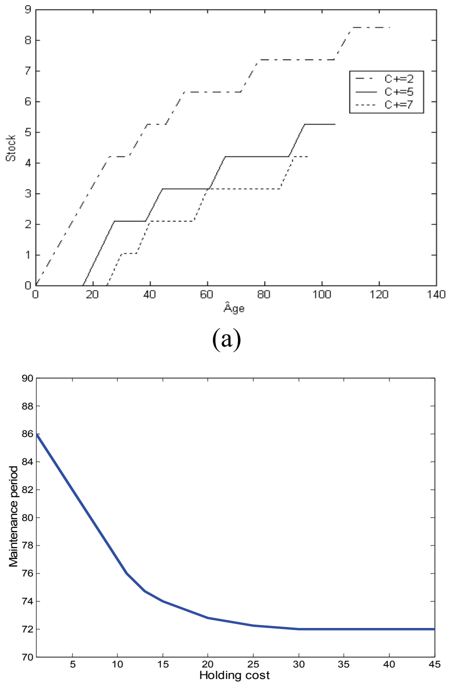
(19)

To illustrate the structure of the MADP, we consider $c_1 = 5000$, $c_2 = 3000$, $c^- = 50$, $c^+ = 2$ and obtain optimal values of S_j and dA_j. The optimal control policy to be applied to the considered manufacturing firm is defined by Equation (13) with the obtained values of S_i^* and dA_i^*, $i = 1, \ldots, 9$. The results presented in this paper indicate that as expected, the optimal production policy for the considered manufacturing system is characterized by two types of parameters, namely, optimal threshold level S_j^* and dA_i^*, $i = 1, \ldots,$ k such that the scheduled preventive maintenance period T^* is given by:

$$T = \sum_{j=1}^{k} dA_j$$

Where k is a random value obtained from the application of the algorithm presented in Figure 6. The control policy (19) is then completely defined by the values of S_j, dA_j and k.

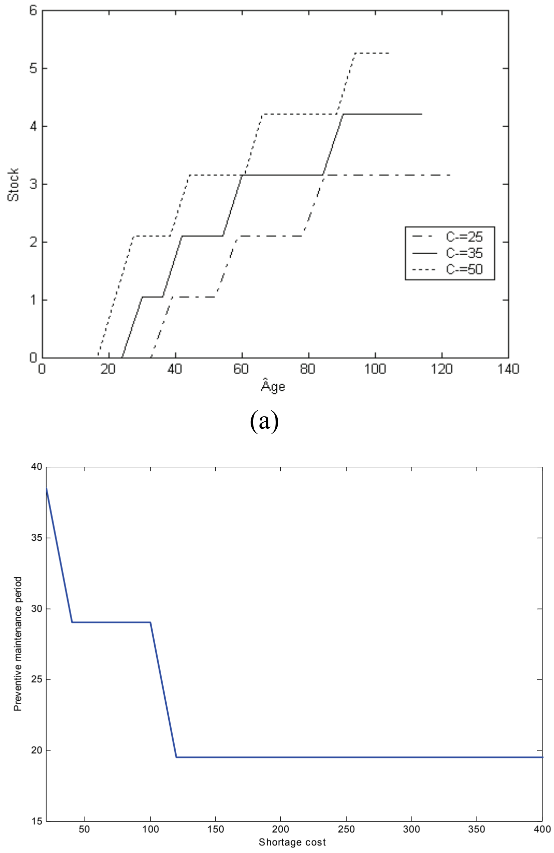
The trends of the curves shown in Figures 7 to 11 confirm the robustness of the proposed approach through a sensitivity analysis. This is performed by threshold levels and scheduled preventive maintenance periods versus an overall incurred cost including inventory, backlog, corrective and preventive costs. The asymptotic behaviour, which is well illustrated in Figures 7 to 11, clearly shows that the results obtained make sense and that the proposed approach is robust.



(a)

(b)

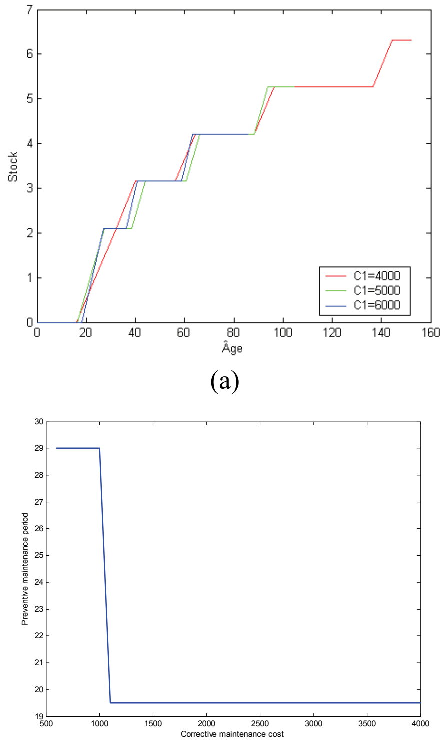
Figure 7: Optimal production and preventive maintenance policies for different inventory costs



(a)

(b)

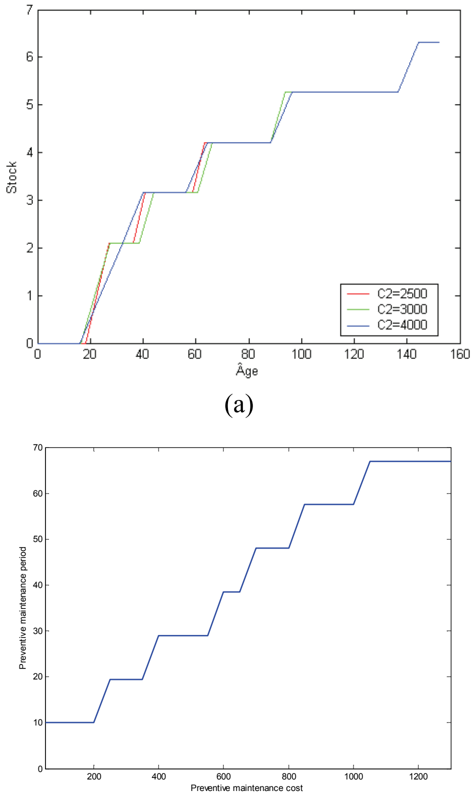
Figure 8: Optimal production and preventive maintenance policies for different backlog costs



(a)

(b)

Figure 9: Optimal production and preventive maintenance policies for different corrective maintenance costs



(a)

(b)

Figure 10: Optimal production and preventive maintenance policies for different preventive maintenance costs

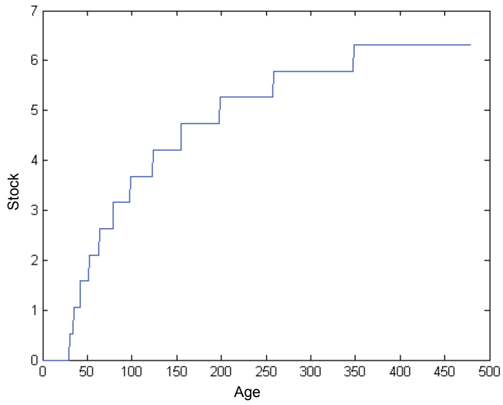


Figure 11: Multiple stage age-dependent hedging point policy with $T \to \infty$

8. CONCLUSION

A production inventory and preventive maintenance system with general characteristics and realistic assumptions has been considered here. From the results obtained, the modeling approach provides a useful tool for studying the effects of various system parameters (inventory, backlog and maintenance costs) on the overall incurred cost, as outlined by the sensitivities analysis. A new concept, called here MADP, is introduced, which is an extension of the modified age-dependent hedging point policy previously proposed in the control literature. Our model can be extended to include large manufacturing systems by generalizing the control policy given by equation (19) and using a simulation and experimental design approach after a parameterization of the control policy. Details on such an approach can be found in [8].

REFERENCES

Ajodhya, N.D., Damodar, A., 2004. Age replacement of components during IFR Delay Time, *IEEE transactions on reliability*, 53(3), 306-312.

Akella, R., Kumar, P.R., 1986. Optimal control of production rate in a ailure-prone manufacturing system. *IEEE Transactions on Automatic Control*, AC-31, 116-126.

Barlow, R.E., Proschan, F., 1965. Mathematical Theory of Reliability, *Wiley, New York*.

Bielecki, T., Kumar, P.R., 1988. Optimality of Zero-Inventory Policies for Unreliable Manufacturing Systems, *Operations Research*, 36, 532-541.

Boukas, E.K., Haurie A., 1990. Manufacturing Flow Control and Preventive Maintenance: A Stochastic Control Approach, *IEEE Trans. on Automatic Control*, Vol. 33, No. 9, 1024-1031.

Boukas, E.K., Kenné, J.P., Zhu, Q., 1995. Age-Dependent Hedging Point Policies in *Manufacturing Systems, American Automatic Control Council,* June 21-23, Seattle (Washington, USA).

Das, T.T., Sarkar, S., 1999. Optimal Preventive Maintenance in a Production Inventory System, *IIE transactions on quality and reliability engineering*, vol. 31, pp. 537-551.

Gharbi, A., Kenné, J.P., 2003. Optimal production control problem in stochastic multiple-product multiple-machine manufacturing systems. *IIE Transactions*, 35, 941-952.

Hong, C, Jionghu, A.J., 2003. Cost-Variability-Sensitive Preventive Maintenance Considering Management Risk, *IIE Transactions,* 35, 1091-1101.

Iravani, S.M., Duenyas, I, 2002. Integrated Maintenance and Production Control of a Deteriorating Production System, *IIE Transactions*, 34, 423-435.

Karen, K.Y., Liu H., Yin G.G., 2003. Stochastic Models and Numerical Solutions for production Planning with Applications to the paper Industry, *Computers and Chemical Engineering,* 27,1693-1706.

Kenné, J.P., Gharbi, A., 1999. Experimental design in production and maintenance control problem of a single machine, single product manufacturing system. *International Journal of Production Research*, 37(3), 621-637.

Kenné, J.P., Boukas, E.K., 2003. Hierarchical Control of Production and Maintenance Rates in Manufacturing Systems, *Journal of Quality in Maintenance Engineering*, 9(1), 66-82.

Cheung, K, L., Hausman, W.H., 1997. Joint determination of preventive maintenance and safety stocks in an unreliable production environment, Naval Research Logistics, 44(3), 257-272.

Rishel, R., 1975. Dynamic Programming and Minimum Principles for Systems With Jump Markov Disturbances, *SIAM Journal on Control*, 13, 338-371.

Salameh, M.K., Ghattas, R.E., 2001. Optimal Just-in-time Buffer Inventory for Regular Preventive Maintenance, *International Journal of Production Economics,* 74(1-3), 157-161.

Savsar, M., 2006. Effects of Maintenance Policies on the Productivity of Flexible Manufacturing Cells, Omega – the international Journal of Management Science, 34, 274-282.

Sheu, S-H., Griffith, W.S., 2001. Optimal Age-Replacement Policy With Age-dependent Minimal-Repair and Random-Leadtime, *IEEE transactions on reliability,* 50(3), 302-309.

INVENTORY MANAGEMENT COSTS ANALYSIS UNDER DIFFERENT CONTROL POLICIES

Antonio Cimino[a], Duilio Curcio[b], Giovanni Mirabelli[c], Enrico Papoff[d]

[a] [b] [c] [d]Modeling & Simulation Center - Laboratory of Enterprise Solutions (MSC – LES)
M&S Net Center at Department of Mechanical Engineering
University of Calabria
Via Pietro Bucci, Rende, 87036, ITALY

[a] [b] [c] [d]{acimino, dcurcio, g.mirabelli, e.papoff}@unical.it

ABSTRACT

The objective of the paper is to use a simulator, implemented in Anylogic™ by *XJ Technologies*, for testing the effect of demand variability, demand intensity and lead time on the inventory management costs within a warehouse.

Keywords: inventory management, costs analysis, Anylogic™

1. INTRODUCTION

Nowadays Inventory Control (*IC*) has a critical role in systems' management: a correct IC allows the manufacturing plant to satisfy customers' demand reducing operation costs; from the other side, IC failures may generate serious consequences, as reported in Lee and Wu (2006).

More in detail, the importance of IC becomes more relevant when applied to the whole Supply Chain (*SC*). For instance, the Bullwhip effect, very frequent along the SC, generates large inventories because of the information distortion amplification as the SC moves to the upstream.

According to Stenger (1996), the effective planning and control of inventories is very difficult in the modern manufacturing systems because the theoretical models adopted, based on several restrictive assumptions, need a great amount of data and lack flexibility.

There is a long history of literature about Inventory Systems (*ISs*).

Lee and Billington (1993) provide a detailed overview on multi-echelon inventories; Cohen and Lee (1988) discuss about the lead time definition for replenishment in a network characterized by a single plant with multiple inputs. Fleischmann et al. (1997) propose a general overview on quantitative approaches for production planning and inventory control systems.

Van der Laan et al. (1996) introduce several approaches based on continuous-review models in which inventory policies parameters are optimized by using queuing theory results. From the other side, Whisler (1967) develop a periodic-review model adopting a dynamic programming approach.

Lee and Wu (2006) make studies on replenishment policies in order to reduce the number of backorder and inventory costs.

Heizer and Render (2001) classify four types of inventory for distinguishing relative costs:

- raw material inventory to reduce suppliers' variability in terms of quality, quantity and delivery time;
- work-in-process inventory for providing production changes;
- maintenance and operating inventory to guarantee the correct running of plants;
- finished products inventory represented by items waiting for shipping.

In addition, several research studies have been made on a specific SC node, as for instance, the warehouse systems. Mason et al. (2003) implement an integrated application for warehouse inventory and transportation management. Chen and Samroengraja (2000) carry out a research study on a warehouse, multi-retailer systems for testing the effectiveness of the two allocation policies adopted and to define their optimal parameters' values. Ahire and Schmidt (1996) analyze the performance of a warehouse, n-retailer system with a continuous review inventory policy.

As before mentioned, the focus of this paper is to present a simulation model for analyzing the performance of a warehouse system in terms of inventory costs under the effect of different demand trends and lead times. In the sequel a brief description of each section of the paper is reported. Section 2 reports the description of the simulation model; Section 3 deals with the inventory policy implemented while an application example is reported in Section 4. Finally, conclusions summarize critical issues and results of the paper.

2. THE SIMULATION MODEL

The simulation model recreates all the most important processes and operations which characterize a warehouse system and it becomes a useful tool for testing and monitoring inventory management costs in function of different inventory policies.

The model is implemented using the commercial package Anylogic™ by *XJ Technologies*.

In particular, for reproducing each process and for increasing model flexibility, different classes have been implemented by using software library objects; in addition, ad-hoc programmed routines implement the logics and rules governing the system.

2.1. The conceptual model

Figure 1 proposes the conceptual model of the warehouse system under analysis.

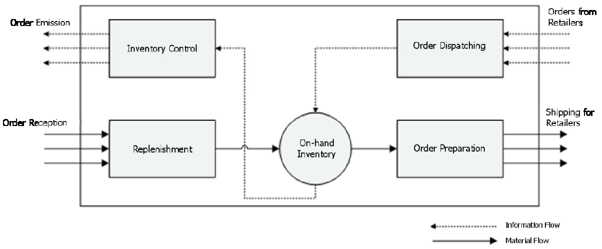


Figure 1: The Warehouse Conceptual Model

As shown in Figure 1, there are two types of data flows:

- the first one related to information flow from retailers;
- the second one concerns items' deliveries from the warehouse to retailers.

In fact, when retailers send orders to the warehouse system, data related to each order are stored in specific databases. After the inventory control, order are processed and items are prepared for deliveries.

2.2. The model setup

Data necessary to model setup are stored in dedicated databases built in Microsoft Excel concerning:

- the trucks arrival time from Suppliers;
- the daily number of trucks from suppliers and for retailers;
- the time interval in which trucks deliver products to the warehouse from Suppliers.
- the shelves levels;
- the number of material handling equipment (i.e. forklifts)
- the shelves capacity
- inventory unitary costs (including manpower)

Such parameters are evaluated before starting the simulation as shown in figure 2.

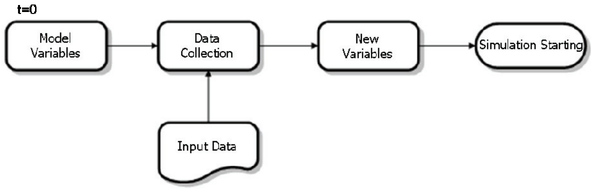


Figure 2: The Model Start-up

The figure 3 shows the orders' processing block diagram.

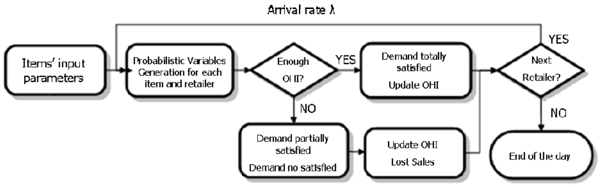


Figure 3: The Orders' Processing Block Diagram

The inter-arrival time Retailers demand is defined by an exponential negative distribution with a λ arrival rate. Each quantity requested is determined by uniform distributions, as reported in Table 1.

Table 1: Probability Distribution for each Retailer

Probability	***Distribution***
P_1	Unif (a_1,b_1)
P_2	Unif (a_2,b_2)
P_n	Unif (a_n,b_n)

P_n is the probability related to retailer demand obtained from the n-th uniform distribution characterized by the two parameters, a_n and b_n. More in detail, each distribution is related to a particular scenario characterized by particular levels of demand intensity and variability and lead times. Next step consists in verifying if the current on-hand inventory (*OHI*) is necessary to satisfy retailers' requests.

Orders can be:

- fully satisfied;
- partially satisfied.

Items number of incoming orders is compared with the OHI. If there is enough OHI, retailers' demand is totally satisfied and an update of the OHI level is made; if demand is partially satisfied or unsatisfied, the OHI is too updated and lost sales are recorded.

Next Section describes the inventory policies implemented in the simulation model for answering to the following questions:

- the time for purchasing order emission;
- the quantity to be ordered.

3. THE INVENTORY POLICIES IMPLEMENTED

The objective of this paper is to compare, by using a Modeling & Simulation based approach, the performance of a warehouse in terms of inventory costs evaluated in function of three classical different inventory policies:

- the periodic-review, order-up-to-level policy *(R, S)*;
- the periodic-review, order-point, order-up-to-level policy *(R, s, S)*;
- the order-point, order-quantity policy *(s, Q)*.

The notation adopted by authors is the following:

- R_i, review period of the item *i*;
- $S_i(t)$, order-up-to-level at time *t* of the item *i*;
- $s_i(t)$, order-point at time *t* of the item *i*;
- $I_i(t)$, on-hand inventory at time *t* of the item *i*;
- $QO_i(t)$, quantity already on order at time *t* of the item *i*;
- $QS_i(t)$, quantity to be shipped at time *t* of the item *i*;
- $Q_i(t)$, quantity to be ordered at time *t* of the item *i*;
- $D_i(t)$, demand at time *t* of the item *i*;
- LT_i, lead time of the item *i*.

3.1. The periodic-review, order-up-to-level policy *(R,S)*

As reported in Silver et al. (1998), this policy is the classical replenishment cycle policy, particularly adopted by companies not using computer control techniques.

This policy works better when all the items are ordered from the same supplier.

At each review period a control on the Inventory Position ($IP_i(t)$) is made and, if necessary, a quantity is ordered to increase it up to the re-order level $S_i(t)$.

$IP_i(t)$ is evaluated as the on-hand inventory plus the quantity already on order minus the quantity to be shipped:

$$IP_i(t) = I_i(t) + QO_i(t) - QS_i(t) \tag{1}$$

In particular, the re-order level $S_i(t)$ is evaluated in this way:

$$S_i(t) = LTD_i(t) + SS_i \tag{2}$$

where:

- $LTD_i(t)$ is the demand forecast over the lead time;
- SS_i is the safety stock calculated as a standard deviation of the lead time demand.

The quantity to be ordered $Q_i(t)$ is given from the following equation:

$$Q_i(t) = S_i(t) - IP_i(t) = LTD_i(t) + SS_i - IP_i(t) \tag{3}$$

This policy is particularly adopted for production systems characterized by a demand pattern changing with time.

3.2. The periodic-review policy *(R,s,S)*

This policy can be considered as a combination of the order-point, order-up-to-level policy *(s,S)* and of the periodic-review, order-up-to-level policy *(R,S)*.

$IP_i(t)$ is checked every review period so two cases can occur:

- $IP_i(t)$ is at or below the re-order point $s_i(t)$;
- $IP_i(t)$ is above $s_i(t)$.

In the first case the quantity to be ordered is enough to raise the $IP_i(t)$ to $S_i(t)$ while in the second nothing is ordered until the next review.

According to Silver et al. (1998), it is demonstrated that, in function of specific assumptions on demand pattern and cost factors involved, the *(R,s,S)* policy generates total costs lower those of other inventory policies.

More in detail, in this policy the re-order point is evaluated according to equation 4:

$$s_i(t) = LT * \frac{\sum_{t}^{t+R-1} D_i(t)}{R_i} + SS_i \tag{4}$$

while:

$$S_i(t) = \frac{\sum_{t}^{t+R-1} D_i(t)}{R_i} + s_i(t) \tag{5}$$

The quantity to be ordered is:

$$Q_i(t) = S_i(t) - IP_i(t) \tag{6}$$

3.3. The order-point, order-quantity policy *(s,Q)*

This policy is a continuous review policy. The fixed quantity $Q_i(t)$ is ordered when the $IP_i(t)$ equals or it is less than the re-order point $s_i(t)$.

In particular, the quantity to be ordered is defined according to the economic order quantity (*EOQ*).

As it is possible to understand this policy is simple to apply and in particular errors that can occur are very low.

4. THE DESIGN OF EXPERIMENTS

As mentioned into the introduction, the goal of this paper is to use the simulator, implemented in Anylogic™ by *XJ Technologies*, for testing warehouse performance under different operative scenarios in order to understand how changes in some critical input parameters (demand variability and intensity and lead time) affect inventory costs monitored in function of three different inventory policies.

The total Inventory management costs are evaluated according to equation 7:

$$TMC_i(t) = TFC_i + UCS_i * AHOI_i(t) + UPC_i * LS_i(t) \qquad (7)$$

where:

- TFC_i are the total fixed costs for the item *i*;
- UCS_i is the unitary storage cost for the item *i*;
- $AHOI_i(t)$ represents the average HOI at time *t* for the item *i*;
- UPC_i is the unitary penalty cost for the item *i*;
- $LS_i(t)$ are the lost sales at time *t* for the item *i*.

Each operative scenario is obtained by changing system input parameters between specific values and conditions. In particular, the input parameters are:

- demand intensity (*IN*) which can assume three different conditions (low, medium, high);
- demand variability (*VAR*) which can vary from low to high conditions;
- lead time (*LT*) which can be changed respectively in one day, three and five days.

Table 2 shows factors and levels adopted for the design of experiments (*DOE*).

Table 2: Factors and Levels of DOE

Factors	***Level 1***	***Level 2***	***Level 3***
IN	Low	Medium	High
VAR	Low	Medium	High
LT	1	3	5

Each factor has three levels: in particular, Level 1 indicates the lowest value for the factor, Level 2 the medium value while Level 3 represents the greatest value.

In order to test all the possible factors combinations, the total number of the simulation runs is 3^3 (3 factors x 3 levels x 3 values). Each simulation run has been replicated three times, so the total number of replications is 27 (27x3=81).

5. SIMULATION RESULTS ANALYSIS

In this Section, results analyses for the total inventory management costs evaluated in function of the different inventory policies implemented are reported.

Table 3 shows all the 27 combinations of the input factors; the first three columns report settings indicating the low, medium and high levels for each factor considered while the last columns contain the total inventory costs results provided by the simulation model for the three inventory policies.

Table 3: Output Data for Total Inventory Management Costs

IN	***VAR***	***LT***	***(R,S)***	***(R,s,S)***	***(s,Q)***
Low	Low	1	141944	149464	148585
Medium	Low	1	334906	350763	342885
High	Low	1	550285	557752	534855
Low	Medium	1	166029	163635	160014
Medium	Medium	1	375278	380820	369176
High	Medium	1	601292	616563	567996
Low	High	1	177830	166757	165775
Medium	High	1	407363	411343	392193
High	High	1	754792	639089	651687
Low	Low	3	154986	159391	152598
Medium	Low	3	360885	365780	386033
High	Low	3	576094	585462	559106
Low	Medium	3	177588	173490	172565
Medium	Medium	3	411387	415806	399984
High	Medium	3	669729	658459	628045
Low	High	3	192845	203489	186893
Medium	High	3	441840	438648	436726
High	High	3	709631	732033	709365
Low	Low	5	163045	168096	157921
Medium	Low	5	370033	383555	382664
High	Low	5	607295	612608	598744
Low	Medium	5	186265	191701	184587
Medium	Medium	5	412897	462012	436579
High	Medium	5	747470	721440	700442
Low	High	5	208494	192065	205423
Medium	High	5	467202	502172	508377
High	High	5	754792	837508	723377

5.1. Simulation results analysis for the total inventory management costs – *demand variability*

These results for total inventory management costs are obtained in function of different operative scenarios. Let us consider the case in which demand variability and lead times are kept constants and the demand intensity is varied.

The total inventory management costs comparison in correspondence of low demand variability and lead time of one day shows that the (R,S) and (R,s,S) have a similar performance but the (s,Q) policy performs better because it provides the lowest inventory management costs.

For low variability demand and lead time of three days, the (s,Q) policy performs better than the others for low and high demand intensity while for medium demand intensity the (R,S) policy gives lower inventory management costs than the others policies. This is confirmed by numerical values reported in Table 3.

In correspondence of a lead time of 5 days, the (R,S) policy performs better for low and medium demand intensity, but the (s,Q) policy provides the lowest total inventory management costs for high demand intensity. The (R,s,S) policy gives the highest costs.

The same analysis is carried out changing variability setting to medium and high. In particular, the (s,Q) policy performs better than the others.

5.2. Simulation results analysis for the total inventory management costs – *demand variability*

In this section the authors describe the results related to demand variability; the total inventory management costs are compared keeping constant both demand intensity and lead times.

The demand intensity is set to its lower value and lead time to 5 days. For low and medium demand variability the (s,Q) policy provides the lowest inventory management costs, but for high variability the better policy is the (R,s,S) policy.

The total inventory management costs trend obtained keeping constant demand intensity to its medium value and lead time to 3 days is analyzed. For low demand variability, the (R,S) policy provides the lowest inventory management costs, but for medium and high variability the better policy is the (s,Q) policy.

Keeping fixed intensity to its high setting and lead time to one day, three cases occur:

- for low demand variability the better policy is the (R,S) policy;
- for medium demand variability the better policy is the (s,Q) policy;
- for high demand variability the better policy is the (R,s,S) policy.

5.3. Simulation results analysis for the total inventory management costs – *lead time*

The same analyses have been carried out taking into consideration lead times values and all the combinations of demand intensity and variability. More in detail, in this section, authors presents results related to:

- low demand intensity and variability;
- medium demand intensity and variability;
- high demand intensity and variability.

The total inventory management costs of the three different inventory policies implemented are compared. For low demand intensity and variability, these cases occur:

- for lead time of one day, the (R,S) policy provides the lowest total inventory management costs;
- for lead time of three days, the (s,Q) policy performs better;
- for lead time of five days, the (s,Q) policy guarantees the lowest total inventory management costs.

Figure 12: Total Inventory Management Costs – Low Intensity and Low Variability

For medium demand intensity and variability and for lead time values of one day and 3 days, the three policies provide approximately the same total costs, but for lead time of five days the (R,S) policy provides the lowest inventory management costs.

The total inventory management costs trend for high demand intensity and variability is as follows: the (R,S) policy is the worst policy for lead time of one day; for lead time of three days, the three inventory policies provide approximately the same total inventory management costs while for lead time of 5 days, the best policy is the (s,Q) policy because it provides the smallest total inventory management costs.

6. CONCLUSIONS

In this paper a simulation model of a warehouse system has been developed. The software tool adopted for the model implementation is Anylogic™ by *XJ Technologies*. The goal of the research work consists in testing system behavior under different operative scenarios. The performance parameter chosen is represented by the total inventory management costs evaluated according to three different inventory policies, the *(R,S)*, the *(R,s,S)* and the *(s,Q)*.

The output data of the simulation model allows to understand how demand variability and intensity and lead time affect the total inventory management costs.

Results obtained highlight the weight of each factor in the performance parameter evaluation allowing to choose the best inventory management policy.

REFERENCES

Ahire, S. L., Schmidt, C. P., 1996. A model for a Mixed Continuous-Periodic Review One-Warehouse, N-Retailer inventory system. *Operational Research*, 92, 69 – 82.

Amato, F., Basile, F., Carbone, C., Chiacchio, P., 2005. An Approach to control automated warehouse systems. *Control Engineering Practice*, 13, 1223–1241.

Axsäter, S. 2003. Optimal Policies for Serial Inventory Systems Under Fill Rate Constraints. *Management Science*, 49(2), 247-253.

Baganha, M. P., Cohen, M.,1998. The stabilizing effect of inventory in supply chains. *Operations Research* 46 (3), S72-S73.

Balci, O., 1998. Verification, Validation and Testing, in *Handbook of Simulation*, edited by J. Banks, pp. 335-393, Wiley Interscience: New York.

Ballou R., 2003. *Business Logistics/Supply Chain Management*. Prentice Hall, Readings.

Beamon, B. M. 1999. Measuring supply chain performance.*International Journal of Operations and Production Management* 19 (3), 275-292.

Bertsimas, D., Thiele, A., 2006. A robust Optimization approach to Inventory Theory, *Oper. Res.*, 54(1), 150 – 168.

Birkin, M., Clarke, G., Clarke, M.P, 2002. *Retail Intelligence and Network Planning*. New York: J.Wiley.

Blackwell, R.D., 1997. *From Mind to Market*. New York: Harper-Collins Publisher.

Brunetta, L., Romanin-Jacur, G., 1999. Passenger and baggage flow in an airport terminal: a flexible simulation model. *Air Traffic Management*, 6, 361 – 363.

Bruzzone A.G., Simone S., Bocca E., 2004. Intelligent Management of a Logistics Platform for Fresh Goods. *Proceedings of SCI2004, Orlando, July.*

Bruzzone, A.G., Longo, F., Massei, M., Saetta, S., 2006. The vulnerability of supply chain as key factor in supply chain management. *Proceedings of Summer Computer Simulation Conference* pp. 181-186, Calgary, Canada.

Chan, F.T.S., Chan H.K., 2005. Simulation modeling for comparative evaluation of supply chain management strategies. *Journal of Advanced Manufacturing Technology* 25, 998–1006.

Chang, Y., H. Makatosoris, 2001. Supply chain modelling using simulation, *Int. J. Simulation*, 2001, 2(1), 24 – 30.

Chen, F., Samroengraja, R., 2000. A Staggered Ordering Policy for One-Warehouse, Multiretailer Systems. *Operations Research*, 48 (2), 281 – 293.

Christofides, N., Colloff, I., 1973. The Rearrangement of Items in a Warehouse. *Operations Research*, 21 (2), 577 – 589.

Cohen, M., Lee, H., 1988. Strategic Analysis of integrated production-distribution systems: models and methods. *Operational Research*, 36, 216 – 228.

Cormier, G., Gunn, E.A., 1996. Simple Models and Insights for Warehouse Sizing. *Operational Research*, 47, 690 – 696.

Curcio, D., Longo, F., Mirabelli, G., 2007. Inventory policies comparison in a manufacturing process using modeling & simulation. *Proceedings of the International Mediterranean Modelling Multiconference,* pp. 237 – 242. October 04 – 06, Bergeggi (Italy).

D'Agostino R.B., Stephens, M.A., 1986. *Goodness of fit techniques*, 1986, Marcel Dekker: New York.

Daganzo, C. F., On the stability of the supply chains, *Op. Res.*, 2004, 52(6), 909 – 921.

De Sensi, G., Longo, F., Mirabelli, G., Papoff, E., 2006. Ants colony system for supply chain routes optimization, in the *Harbour, Maritime & Multimodal Logistics Modelling and Simulation Conference*, 763 – 768.

De Sensi, G. , Longo, F., Mirabelli, G., 2007. Inventory policies analysis under demand patterns and lead times constraints in a real supply chain. *International Journal of Production Research*, doi: 10.1080/00207540701528776

Department of Defense, Deputy under Secretary of Defence, DoD modelling and simulation (M&S) management, *DoD Directive 5000.59*, 1994

Dunn R.H., 1987. The quest for software reliability, in *Handbook of software quality assurance*, edited by G.G. Schulmeyer and J.I. McManus, pp. 342 – 384, Van Nostrand Reynold: New york.

Eben-Chaime, M., Pliskin, N., Sosna, D., 1997. Operations management of multiple machine automatic warehousing systems. *Production Economics*, 51, 83–98.

Eben-Chaime, M., Pliskin, N., Sosna, D., 2004. An integrated architecture for simulation. *Computer and Industrial Engineering*, 46, 159–170.

Fleisch, E., Tellkamp, C., 2005. Inventory inaccuracy and supply chain performance: a simulation study of a retail supply chain. *International Journal of Production Economics* 95, 373–385.

Fleischmann, M., Bloemhof-Ruwaard, J.M., Dekker, R., Van der Laan, E., Van Nunen, J.A.E.E., Va Wassenhove, L.N., 1997. Quantitative models for reverse logistics: a Review. *Operational Research*, 103, 1 – 17.

Ganeshan, R., Boone, T., Stenger, A. J., 2001. The impact of inventory and how planning parameters on supply chain performance: An exploratory study. *International Journal of Production Economics* 71, 111–118.

Giannoccaro, I., Pontrandolfo, P., 2002. Inventory management in supply chains: a reinforcement learning approach, *Int. J. Prod. Econ.*, 2002, 78, 153 – 161.

Graves, S.C., Willems, S. P., 2005. Optimizing the Supply Chain Configuration for New Product. *Management Science*, 51(8), 1165–1180.

Heizer, J. and B. Render. 2001. *Operations Management.*6th ed. N.J.: Upper Saddle River.

Hill, R.M., 1989. Allocating Warehouse Stock in a Retail Chain. *Operational Research*, 40 (11), 983 – 992.

Ingalls, R. G., 1998. The value of simulation in modeling supply chain. *Proceedings of the 1998 Winter Simulation Conference,* pp. 1371-1375 Washington D.C.

Johonson N.L., Kotz, S., Balakrishnan, N., 1994. *Continuous Univariate Distributions*, Vol. 1, 2nd ed., Houghton Mifflin: Boston.

Johonson N.L., Kotz, S., Balakrishnan, N., 1995. *Continuous Univariate Distributions*, Vol. 2, 2nd ed., Houghton Mifflin: Boston.

Johonson N.L., Kotz, S., Kemp, A.W., 1992. *Univariate Discrete Distributions*, 2nd ed., Houghton Mifflin: Boston.

Kelle, P., Milne, A. The effect of (s, S) ordering policy on the supply chain, *Int. J. Prod. Econ.*, 59, 113 – 122.

Kreipl, S., Pinedo, M., 2004. Planning and scheduling in supply chains: an overview of Issues in Practise, *Prod. Op. Manag.*, 2004, 13(1), 77 – 92.

Hoare, N. P., Beasley, J. E., 2001. Placing Boxes on Shelves: A Case Study. *Operational Research*, 52 (6), 605 – 614.

Lee, C., Çetinkaya, S., Jaruphongsa, W., 2003. A Dynamic Model for Inventory Lot Sizing and Outbound Shipment Scheduling at a Third-Party

Warehouse. *Operations Research*, 51 (5), 735 – 747.

Lee, H.T., Wu, J.C., 2006. A study on inventory replenishment policies in a two-echelon supply chain system. *Computers & Industrial Engineering*, 51, 257 – 263.

Lee, H., Billington, C., 1993. Material Management in decentralized supply chains. *Operational Research*, 41, 835 – 847.

Li, J., Shaw, M., Sikora, R., Tan, G. W., Yang, R., 2001. *The effects of Information sharing strategies on supply chain performance*, Working paper, University of Illinois Urbana-Champaign.

Liebeskind, A., 2005. *How to Optimize Your Warehouse Operations*. 1st ed. USA: Industrial Data & Information Inc.

Longo, F., Mirabelli, G., Papoff, E., 2005. Modeling Analysis and Simulation of a supply chain devoted to support pharmaceutical business retail, *Proceedings of the 18th International Conference on Production Research, July 31 – August 4 , Salerno, Italy.*

Longo, F., Mirabelli, G., Papoff, E., 2006. Material Flow Analysis and Plant Lay-Out Optimization of a Manufacturing System. *International Journal of Computing*, 5(1), 107-116.

Longo, F., Mirabelli, G., 2008. An Advanced Supply Chain Management Tool Based on Modeling & Simulation, *Computer and Industrial Engineering*, Vol 54(3), 570-588.

Moinzadeh, K., 2002. A Multi-Echelon Inventory System with Information Exchange. *Management Science*, 48(3), 414-426.

Monks, J.G., 1996. *Operations Management*, New york: McGraw Hill.

Montgomery, C.D. and Runger, G.C., 2003. *Applied Statistics and Probability for Engineers*. 3rd ed. USA: John Wiley & Sons.

Mulcahy, D.E., 1993. *Warehouse Distribution and Operations Handbook*. 1st ed. USA: McGraw-Hill.

Nahmias, S., 1982. Perishable Inventory Theory: A Review. *Operations Research*, 30 (4), 680 – 708.

Napolitano, M., Gross, J., 2003. *The Time, Space & Cost Guide to Better Warehouse Design*. 2nd ed. USA: Distribution Group.

O'Leary, D.E., 2000. *Enterprise Resource Planning Systems: Systems, Life Cycle, Electronic Commerce, and Risk*. Cambridge University Press.

Parker, R.P., Kapuscinski, R., 2004. Optimal Policies for a Capacitated Two-Echelon Inventory system, *Op. Res.*, 2004, 52(5), 739 – 755.

Persson, F. , Olhager, J., 2002. Performance simulation of supply chain designs, *International Journal of Production Economics*, 77, 231-245.

Ratliff, H.D., Rosenthal, A.S., 1983. Orderpicking in a rectangular warehouse: a solvable case of traveling salesman problem. *Operations Research*, 31, 507 – 521.

Roberson, J.F, 1994. *Logistics Handbook*. Free Press.

Roundy, R. O., Muckstadt, J. A., 2000. Heuristic computation of periodic-review base stock inventory policies, *Manag. Sc.,* 46(1), 104 – 109.

Sanjay, J., 2004. Supply chain management tradeoffs analysis, *Proceedings of the Winter Simulation Conference*, pp. 1358-1364. Washington (USA).

Seifert, D., Marzian, R., McLaughlin, J., 2003. *Collaborative Planning, Forecasting, and Replenishment: How to Create a Supply Chain Advantage*. Amacom

Silver, E.A., Pyke, D.F., Peterson, R., 1998. *Inventory Management and Production Planning and Scheduling.* 3rd ed. USA: John Wiley & Sons.

Simchi-Levi, D., Kaminsky, P., Simchi-Levi, E. *Designig and Managing the Supply Chain.* McGraw-Hill.

Stadtler, H., Kilger, C., 2000. *Supply chain management and advanced planning*, Springer: Berlin.

Stenger, A.J., 1996. Reducing inventories in a multi-echelon manufacturing firm: a case study. *Production Economics*, 45, 239 – 249.

Suwanruji P., Enns, S.T., 2006. Evaluating the effects of capacity constraints and demand patterns on supply chain replenishment strategies, *Int. J. of Prod. Res.*, 44(21), pp. 4607 – 4629.

Van der Laan, E., Salomon, M., Dekker, R., 1996. Product remanufacturing and disposal: A numerical comparison of alternative strategies. *Production Economics*, 45, 489 – 498.

Whisler, W.D., 1967. A stochastic inventory model for rented equipment. *Management Science*, 13, 640 – 647.

AUTHORS BIOGRAPHIES

ANTONIO CIMINO was born in Catanzaro (Italy) in October the 1th, 1983. He took his degree in Management Engineering, summa cum Laude, in September 2007 from the University of Calabria. He is currently PhD student at the Mechanical Department of University of Calabria. His research activities concern the integration of ergonomic standards, work measurement techniques, artificial intelligence techniques and Modeling & Simulation tools for the effective workplace design. He collaborates with the Industrial Engineering Section of the University of Calabria to research projects for supporting innovation technology in SMEs.

DUILIO CURCIO was born in Vibo Valentia (Italy), on December the 15th, 1981. He took the degree in Mechanical Engineering from University of Calabria (2006). He is currently PhD student at the Mechanical Department of University of Calabria. His research activities include Modeling & Simulation and Inventory Management theory for production systems and Supply Chain design and management. He collaborates with the Industrial Engineering Section of the University of

Calabria to research projects for supporting Research and Development in SMEs. His e-mail address is <dcurcio@unical.it>.

GIOVANNI MIRABELLI was born in Rende (Italy), on January the 24th, 1963. He took the degree in Industrial Technology Engineering from University of Calabria. He is currently researcher at the Mechanical Department (Industrial Engineering Section) of University of Calabria. His research interests regard work measurement and human reliability. His e-mail address is <g.mirabelli@unical.it>.

ENRICO PAPOFF was born in Naples (Italy) on February the 03rd, 1948. He took the degree in Mechanical Engineering from University of Napoli Federico II, in 1973. He is currently Associate Professor at the Mechanical Department (Industrial Engineering Section) of the University of Calabria. His research interests regard project management and business plans. His e-mail address is <e.papoff@unical.it>.

DEMAND FORECASTING AND LOT SIZING HEURISTICS TO GENERATE COST-EFFECTIVE PRODUCTION PLANS: A SIMULATION STUDY ON A COMPANY IN THE WOOD FLOORS SECTOR

Lorenzo Tiacci[(a)], **Stefano Saetta**[(b)]

Università degli Studi di Perugia
Dipartimento di Ingegneria Industriale
Via Duranti, 67 – 06125 Perugia

[(a)]lorenzo.tiacci@unipg.it, [(b)]stefano.saetta@unipg.it

ABSTRACT

Master Production Scheduling (MPS) is a very important activity in manufacturing planning and control because the quality of the MPS can significantly influence the total cost. Unfortunately, many companies do not know their future demands and have to rely on demand forecasts to make production planning decisions. Thus in many cases companies must first select a good forecasting method, and then use forecasted demand as input to the planning phase. The paper presents a simulation study, conducted on an industrial case, of these two phases: selection of the most appropriate forecasting method, and production planning through forecasted demands. The aim is double: to quantify improvements, in terms of total costs decrease, with respect to the actual company policy; to evaluate the impact on total costs of the demand forecasting method inaccuracy.

Keywords: Master Production Scheduling, lot sizing, demand forecasting, inaccuracy.

1. INTRODUCTION

The paper presents an industrial case of a company in the wood floors sector of lot sizing under demand uncertainty.

Numerous researches have studied lot sizing and setup scheduling problems, with reviews by Drexl and Kimms (1997) and Karimi, Fatemi Ghomi and Wilson (2003). Many of them assume deterministic demand, and measure performances of different production planning algorithms and procedures in terms of minimization of total costs and computational time. However in many real cases future demand is unknown, and master production scheduling is based on demand forecasts rather than on actual demand.

In this study we simulate master production scheduling activities, performed by a lot sizing heuristic, using data originated by a forecasting procedure. Outputs from the simulation study are used both to evaluate improvements with respect to the actual company policy, and to evaluate the impact of demand forecasting inaccuracy on total costs.

The simulation study considers the most 5 representative items in terms of sales. The product is a two layers parquet. The top layer is constituted from noble wood, with thickness between 3.5 and 5 mm. The inferior layer, technologically more complex, has a support function and allows keeping the pavement without deformation, contrasting the natural tendency to the movement; it consists in a multilayer cross-sectional fiber that guarantees a final product not deformable.

To investigate the impact of demand forecasting method on total costs through computer simulation, we must obtain the demand forecasts. Two alternative approaches have been used to produce the forecasts in previous studies. One approach is to generate the forecasting error according to some probability distribution and add it to the actual demand, as for example in Lee and Adam (1986). The other is to use a forecasting model to make forecasts based on previous demand (see Zhao, Goodale and Lee 1995; Xie, Lee and Zhao, 2004). In this paper we follow this second approach.

For each item demand data on monthly base, for years 2003-2007, are available.

Different demand forecasting methods (see next paragraph) are applied to each item. The first two years (2003-2004) are used for initialization. Then, each method is applied to obtain forecasts (always on a monthly base) for years 2005-2006 and to calculate per period forecasts errors. The method obtaining the minimum Mean Absolute Deviation for each item is selected to perform forecasts for year 2007.

Note that the forecast models are tested with an entirely different set of data (year 2005-2006) from which the model is formed (2003-2004)

Monthly forecasted demands for year 2007 are then used as input for the production planning algorithm. The production planning phase is performed each month, with a rolling time horizon of 4 months. This length is imposed by the lead time needed to obtain the inferior layer needed to manufacture the final product. If the inferior layer is available, requirements for a periods can be covered by the production planned for the same period; in effect, the application of the noble wood layer can be performed in a relative small amount of time.

The simulation study consisted in generating production plans for each months of year 2007. Outputs (inventory levels, setup costs, holding costs) have been compared to what has been done by the company during the same period.

In order to evaluate the impact of demand forecasting methods inaccuracy on total costs, a Mixed Integer Programming model has been implemented to solve to optimality the lot sizing problem under deterministic demand (that is, the actual demand of year 2007). By comparing the optimal solution (obtained with actual demand), and the one carried out by using forecasted demand, it is possible to quantify the influence of using a certain method to obtain forecasts.

1.1. Notation

t = months index
A = setup cost [€]
v = item value [€]
r_y = annual carrying rate [€/€/year]
$r = r_y/12$ = monthly carrying rate [€/€/month]
d_t = actual demand for period t
$F_{t,j}$ = forecasted demand at period t for period $t+j$
I_t = inventory level at the end of period t
Q_t = quantity produced for period t

The index indicating items is omitted because they are considered independently from each others.

2. DEMAND FORECASTING

Four demand forecasting methods have been tested:
- Moving Average.
- Simple Exponential Smoothing.
- Trend Corrected Exponential Smoothing.
- Trend and Seasonality Corrected Exponential Smoothing.

Each of the above method has been implemented using different values of its characterizing parameters in order to find values that, for each item, minimize the MAD during 2005-2006. Four values for the number of preceding periods (N=2, N=3, N=4 and N=5) have been tested for moving averages, while as far as exponential smoothing methods are concerned, a search procedure has been performed in order to find optimal values for the smoothing parameters α, β and γ.

Figure 1 summarizes the results obtained applying each forecasting method to each demand item historical data, measuring the MAD in years 2005-2006. It can be observed that in all cases the Trend and Seasonality Corrected Exponential Smoothing provides the smaller error, and it is therefore selected to forecast demand for year 2007.

In particular, the values that provide better result, in terms of minimization of MAD, are respectively: α=0.3, β=0.1 and γ=0.1 for item#1; α=0.05, β=0.1 and γ=0.2 for item#2; α=0.05, β=0.01 and γ=0.5 for item #3; α=0.25, β=0.2 and γ=0.1 for item #4; α=0.001, β=0.2 and γ=0.3 for item #5.

Figures 2 to 6 shows, for each item, the actual demand against the forecasted demand obtained by applying the selected method for year 2007.

MAD				
Method / **Article**	Moving Average	Simple Exponential Smoothing	Trend Corrected Exponential Smoothing	Trend and Seasonality Corrected Exponential Smoothing
item #1	≈970	952	954	**874**
item #2	≈210	180	165	**160**
item #3	≈1110	798	815	**722**
item #4	≈550	690	665	**545**
item #5	≈530	517	497	**374**

Figure 1. MAD of different demand forecasting methods on 2002-2006

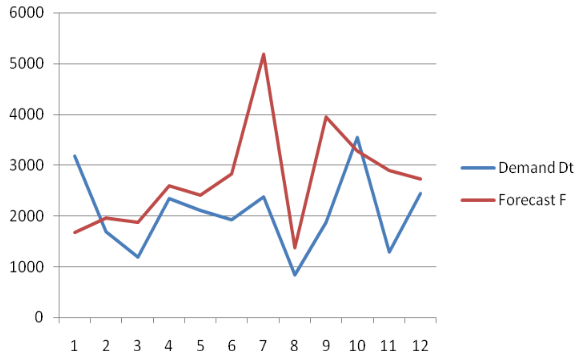


Figure 2. Item#1: comparison between actual and forecasted demand during 2007.

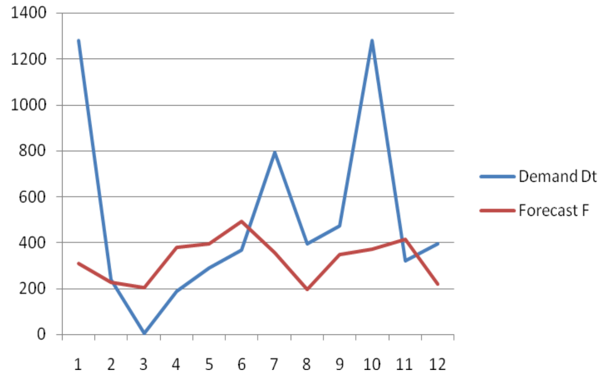


Figure 3. Item#2: comparison between actual and forecasted demand during 2007.

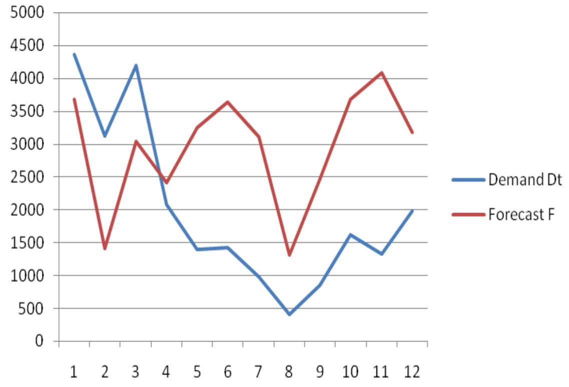


Figure 4. Item#3: comparison between actual and forecasted demand during 2007.

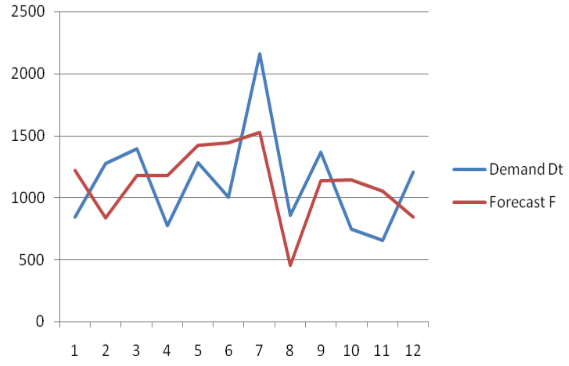


Figure 5. Item#4: comparison between actual and forecasted demand during 2007.

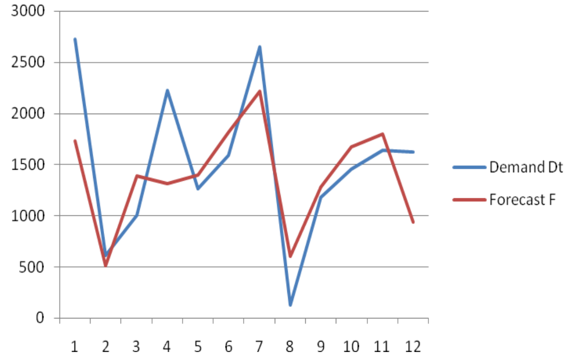


Figure 6. Item#5: comparison between actual and forecasted demand during 2007.

3. PRODUCTION PLANNING

The production planning phase is performed each month, with a rolling time horizon of 4 months. This length is imposed by the lead time needed to obtain the inferior layer needed to manufacture the final product. If the inferior layer is available, requirements for a periods can be covered by the production planned for the same period; in effect the application of the noble wood layer can be performed in a relative small amount of time.

Starting for example by January 2007 (t = 1), the related production plan for this month will indicate production quantities that have to be delivered in January, February, March and April. To do this we need demand requirements for the same periods. Demand requirements for the first month, in this case January, are equal to orders already collected (d_1) and do not need to be forecasted. On the contrary, demand requirements for February, March and April have to be forecasted ($F_{1,1}$, $F_{1,2}$, $F_{1,3}$). We obtain these forecasts using, as mentioned in the previous paragraph, the Trend and Seasonality Corrected Exponential Smoothing:

$$F_{t,j} = (L_t + j \cdot T_t) S_{t+j} \tag{1}$$

where L_t and T_t, are respectively level and trend calculated at time t, and S_{t+j} is the seasonal factor estimated for period $t + j$.

The four data, namely d_1, $F_{1,1}$, $F_{1,2}$, $F_{1,3}$, are used by the planning algorithm to generate (trying to minimize the sum of setup and holding costs, see Paragraph 3.1) productions requirements for periods t = 1, 2, 3 ,4. However, only the first period is implemented, while period 2, 3 and 4 allow estimating raw material requirements. In particular period 4 will indicate the quantity of a new raw material requirement, while period 2 and 3 will be used to correct and refine previous estimates. Naturally, safety stocks of raw materials have to be provided to allow for uncertainty.

The procedure is then repeated in the next month, February (t = 2), and so on. Table 1 shows orders collected for the current month (d_t) and forecasts for the next three months ($F_{t,1}$, $F_{t,2}$, $F_{t,3}$) in each period t for item#1 during 2007. Each row represents the input to the planning algorithm for the current month. Note that forecasts for succeeding periods differ, of course, from orders that will be actually collected.

Table 1 reports all other inputs required by the algorithm to generate the production plan, that is: the initial inventory level, the setup cost, the item value and annual carrying rate.

3.1. Production planning heuristics

The planning algorithm is an adapted version of the Silver Meal heuristics (Silver and Meal 1973), that is usually adopted for deterministic lot sizing. When requirements are deterministic, the heuristic calculates the exact number of periods that have to be covered by a replenishment. For example, if we are in period 1, and T is the number of period covered by a replenishment, the ordered quantity will be equal to:

$$Q(T) = \sum_{j=1}^{T} d_j \tag{2}$$

Table 1. Demand forecasting during the simulation study for Item #1.

	d_t	$F_{t,1}$	$F_{t,2}$	$F_{t,3}$
January	3167	2509	2424	3402
February	1692	2127	2907	2650
March	1188	2507	2272	2619
April	2346	2258	2635	4777
May	2111	2589	4701	1241
June	1926	4319	1134	3215
July	2369	976	2750	2250
August	839	2665	2205	1942
September	1866	2001	1758	1645
October	3541	2217	2122	1805
November	1282	1804	1496	1597
December	2437	1685	1832	1753
	Initial Inv.	A	v	r_y
	3623	296	32	10.00%

The heuristic tries to chose the number of periods *T* (and the associated quantity of replenishment) that minimize future total costs in the time unit. If we indicate with *TRC*(*T*) the total relevant costs (setup costs + holding costs) associated to a replenishment that covers *T* periods, then the total relevant costs per unit time (*TRCUT*(*T*)) associated to the same replenishment are equal to:

$$TRCUT(T) = \frac{TRC(T)}{T} = \frac{C_r + C_c(T)}{T} \tag{3}$$

The heuristic evaluates the *TRCUT*(*T*) for increasing values of *T* (starting from *T* = 1) and when the following condition is satisfied:

$$TRCUT(T+1) > TRCUT(T) \tag{4}$$

T is chosen as the number of periods. In this way, due to the deterministic nature of demand, inventory lasts always an integer number of periods, and each replenishment occurs exactly when the inventory level is equal to 0.

In the non deterministic case, on the contrary, we have to use forecasts instead of actual demand; so it is not guaranteed that ordered quantities will last an integer number of periods.

Thus, if for example we are in period *t* (we are planning to fulfil requirements for periods *t, t+1, t+2, t+3*), first we have to check the inventory level: if $I_{t-1} \geq d_t$ stocks are enough to satisfy requirements for the next month; thus $Q_t = 0$. If $d_t > I_{t-1}$, stocks on hand are not sufficient, and we have to order at least the quantity $Q_t = I_{t-1} - d_t$ to cover the next period requirement. In general, the order quantity $Q_t(T)$ that cover the expected requirements for the next *T* periods will be equal to:

$$Q_t(T) = \begin{cases} I_{t-1} - d_1 & \text{for } T = 1 \\ I_{t-1} - d_1 + \sum_{j=1}^{T-1} F_{t,j} & \text{for } T > 1 \end{cases} \tag{5}$$

Similarly to the deterministic case, the heuristic will decide the number of periods *T* that minimizes the Expected Total Relevant Cost per Unit Time (*ETRCUT*). If *T* = 1:

$$ETRCUT(1) = \frac{A}{1} \tag{6}$$

there are not carrying costs, because we only replenish enough to cover the actual requirements of the first next period.

With *T* = 2 the expected carrying costs are $vrF_{t,1}$, that is the cost of carrying the expected requirement $F_{t,1}$ for one period:

$$ETRCUT(2) = \frac{A + vrF_{t,1}}{2} \tag{7}$$

With *T* = 3 we still carry $F_{t,1}$ for one period, but now we also carry $F_{t,2}$ for two periods. Therefore,

$$ETRCUT(3) = \frac{A + vrF_{t,1} + 2vrF_{t,2}}{3} \tag{8}$$

$$ETRCUT(4) = \frac{A + vrF_{t,1} + 2vrF_{t,2} + 3vrF_{t,3}}{4} \tag{9}$$

As before, we will chose to cover a number of period *T* for which:

$$ETRCUT(T+1) > ETRCUT(T) \tag{10}$$

corresponding to an ordered quantity expressed by the (5). In our particular case, due to the short rolling horizon of 4 months and the limited number of items, all possible values of *ETRCUT* (*T* = 1, 2, 3, 4) have been calculated through (6)(7)(8)(9) and the *T* value that gives the minimum have been chosen.

4. RESULTS

The algorithm has been implemented in Java with the compiler NetBeans IDE6.0.

Table 2 shows the outputs of the simulation study for item#1, consisting in production quantities during 2007 and the corresponding inventory level. Results are compared to what obtained in the same period by the company.

Figures 7 and Figures 8 shows graphically the same comparison for all the 5 items during year 2007. Starting from the same inventory level, the algorithm tends initially not to produce, because stocks on hand are still enough to cover the next periods requirements.

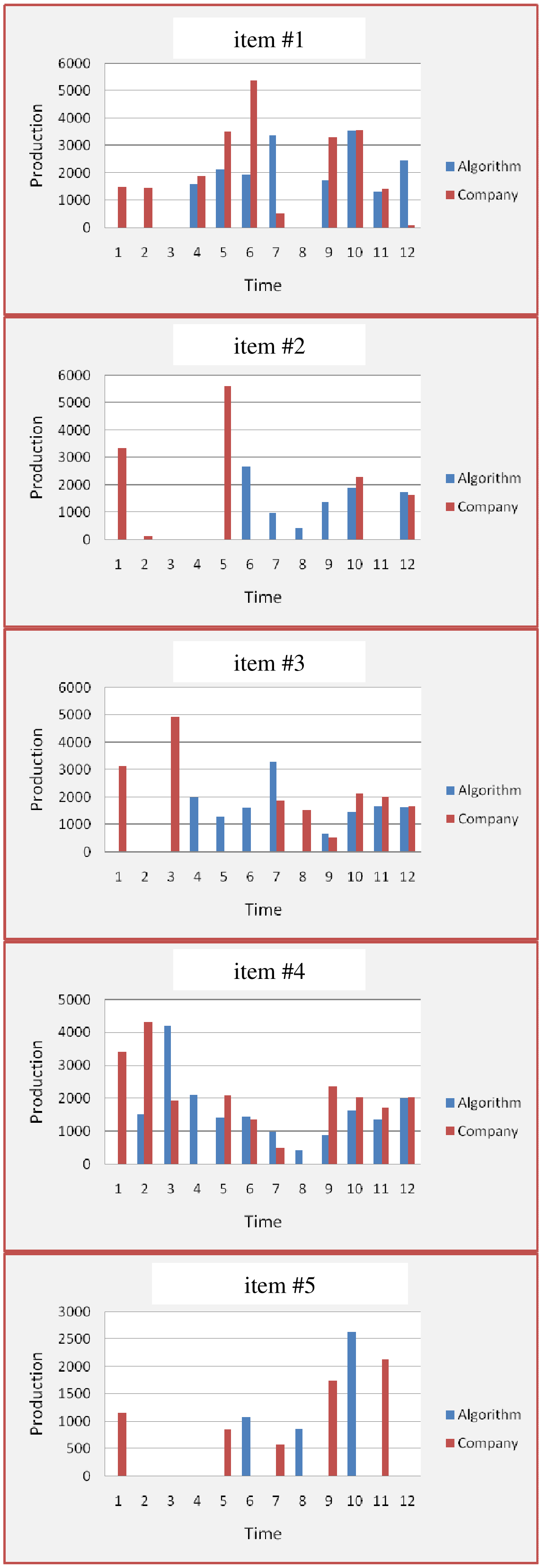


Figure 7. Production quantities: comparison between algorithm and company during year 2007

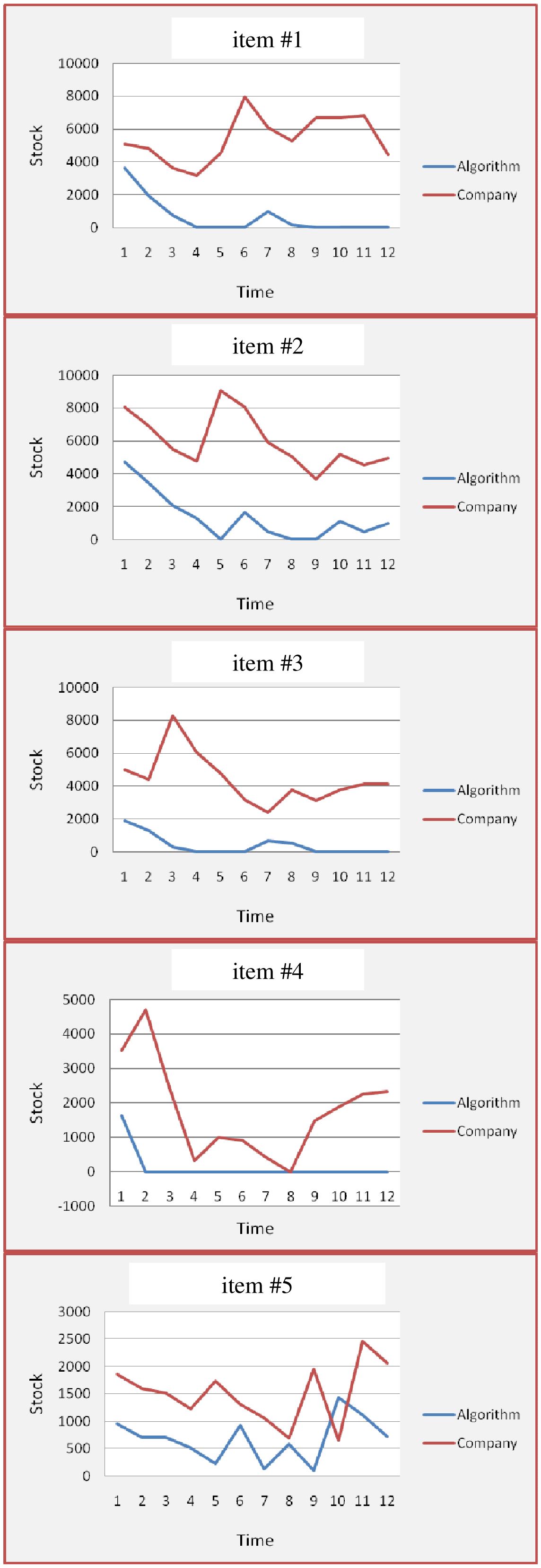


Figure 8: Inventory levels: comparison between algorithm and company during year 2007.

Table 2. Output of the simulation study for item#1

Algorithm		Company	
Production	Inventory	Production	Inventory
0	3633	1469	5102
0	1941	1427	4837
0	753	0	3649
1593	0	1874	3177
2111	0	3486	4552
1926	0	5360	7986
3345	976	493	6110
0	137	0	5271
1729	0	3286	6691
3541	0	3571	6721
1282	0	1391	6831
2437	0	88	4482
N Setup	Average Inv.	N Setup	Average Inv.
8	620	10	5451
Setup Cost	Holding Cost	Setup Cost	Holding Cost
2369	1984	2961	17443
Total Cost		Total Cost	
4352		20403	

On the contrary the company tends always to saturate its capacity, and produces even when it is not still required. Only when the inventory level is much lowered, the algorithm gives production quantities that are similar to the company's ones.

This implies that the number of setups performed by the algorithm is about the same of those performed by the company, but average inventory levels are much lower. The same behaviour is observable from data reported in Table 3 and Figure 9, that show the aggregate results for all 5 five products.

Table 3. Aggregate results of the simulation study.

Algorithm		Company	
Production	Inventory	Production	Inventory
0	15046	12492	25790
1503	9194	5850	24259
4195	5183	6814	22791
5646	2867	1874	16614
4774	1260	13088	23272
8684	3396	6716	23384
8572	2451	4583	18451
1662	1467	1503	17328
4919	301	7895	19129
11099	2554	9980	20254
4771	1871	7221	22022
8048	1890	5379	19398
N Setup	Average Inv.	N Setup	Average Inv.
36	3957	38	21058
Setup Cost	Holding Cost	Setup Cost	Holding Cost
13888	14804	14184	83877
Total Cost		Total Cost	
28692		98061	

The reduction of total costs is impressive (about 70%) and substantially due to the drastic decrease of inventories. It is noteworthy that the algorithm never generates peaks of production quantities higher than those planned by the company.

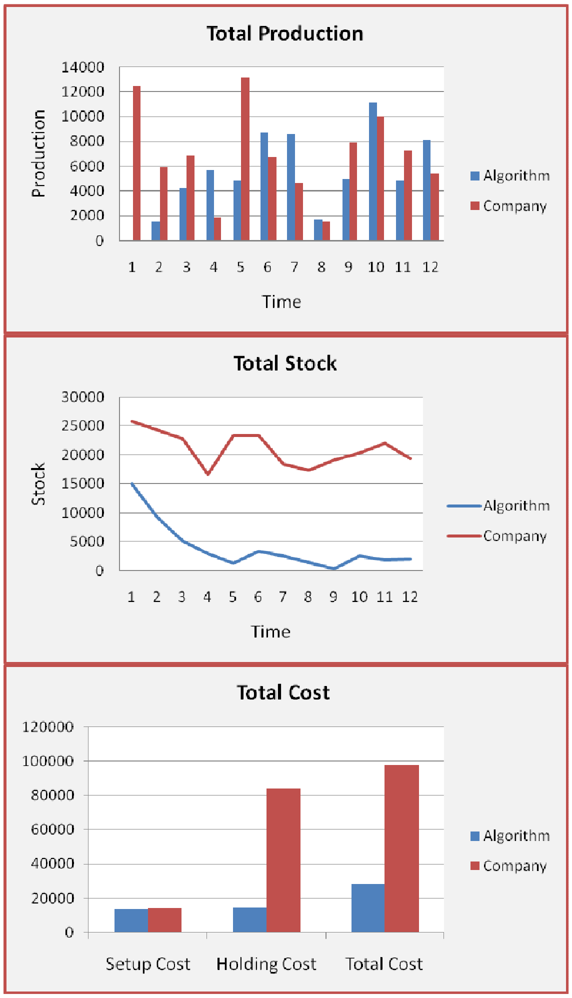


Figure 9. Aggregate results of the simulatio study.

4.1. The impact of uncertainty

In order to evaluate the impact of uncertainty, we evaluate the extra-cost associated to the use of forecasted demand (and the described algorithm) with respect to the optimal solution obtained by considering the actual demand.

In effect, if demand is deterministic, the lot sizing problem can be solved to optimality through the well known dynamic programming approach of Wagner-Whitin (Wagner and Whitin 1958), or through its equivalent formulation as Mixed Integer Programming model.

We consider the following MIP model for each of the five items:

$$Minimize \qquad \sum_{t=1}^{12}\left(vrI_t + Ay_t\right) \qquad (11)$$

subject to:

$$I_{t-1} + Q_t - I_t = d_t \qquad \forall t \qquad (12)$$

$$Q_t \le M\, y_t \qquad \forall t \qquad (13)$$

$$y_t \in \{0,1\} \qquad \forall t \qquad (14)$$

$$Q_t \ge 0 \qquad \forall t \qquad (15)$$

$$I_t \ge 0 \qquad \forall t \qquad (16)$$

where: M is a large positive number (higher than the largest quantity that can be produced in one period), y_t is a binary decision variable equal to 1 if the item is produced for period t and equal to 0 otherwise; I_0 is the initial inventory level. Note that the planning horizon is equal to 12 months and that only actual demand data d_t are considered. The MIP problem has been coded and solved using Xpress® (Dash Optimization Xpress-MP).

Figure 10 shows the comparison between the solution obtained by solving the MIP problem (using actual demand data and a time horizon of 12 months), and the one, previously described, obtained by the algorithm (using forecasted demand data and a rolling horizon of 4 months).

Production quantities are very similar, and also inventory levels show the same behaviour (a consistent initial decrease due to very little production quantities). Nevertheless the gap is still consistent. This is due to fact that just a little difference between forecasted and actual demand causes the impossibility to place orders exactly when the items inventory level are equal to 0, forcing in this way to make a higher number of setups and to have always higher inventory levels. Furthermore, holding costs also increase, when using forecasted data, due to demand overestimate for some of the items.

Table 4 shows costs percentage deviation of optimal and algorithm solutions from the company solution. The difference in total saving between the 70.7% obtained by the algorithm and the 82.3% obtained in the deterministic case can thus be ascribed to the use of forecasted demand data. This gap could be taken as performance indicator of the forecasting method.

Table 4. Percentage deviation of algorithm and optimal solution costs from company costs in 2007

Costs and percentage deviation from Company solution						
	Setup		Holding		Total	
Comp.	14184	/	83877	/	98061	/
Alg.	13888	-2.1%	14804	-82.4%	28692	-70.7%
Opt.	7715	-45.6%	9607	-88.5%	17322	-82.3%

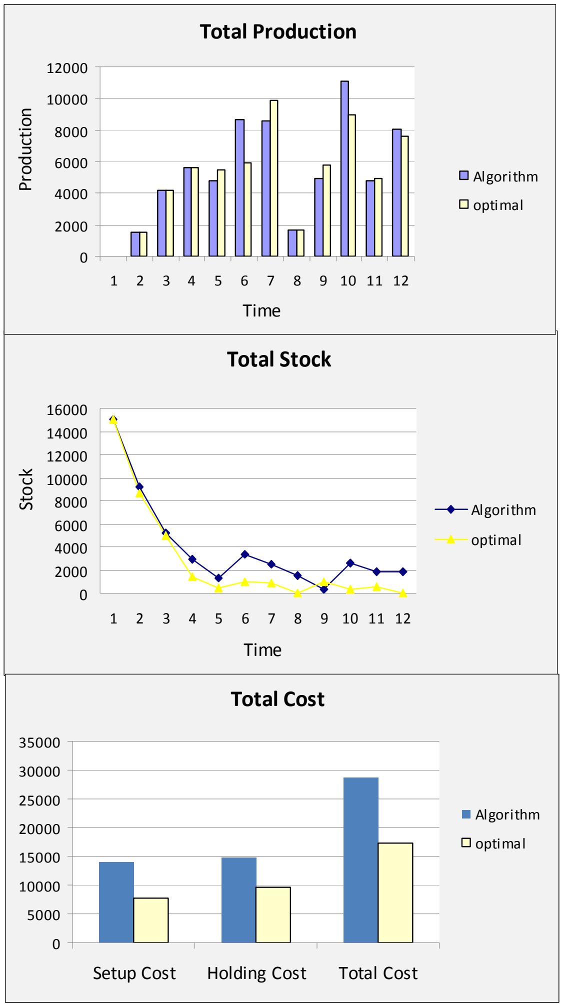


Figure 10. Comparison between optimal and algorithm's solutions.

5. CONSLUSIONS

The model that has been implemented allowed to simulate both the demand forecasting and the production planning activities of the company for one year.

To consider this two phases at the same time in a simulation study is the best way to quantify possible improvements deriving from the adoption of a planning algorithm that, in practice, receives forecasted demand data as inputs.

The procedure allows also to evaluate the goodness of the demand forecasting method not only through per period forecasts error indicators (as the Mean Absolute Deviation) but also through their direct impact on total costs.

Final results show that the application of the selected demand forecasting method and the production planning algorithm allows a total costs reduction, with respect to the actual company policy, up to the 70% in one year.

If actual demand data, instead of forecasts, are considered during the same period, then the planning problem can be solved to optimality. Thus, the impact of the demand forecasting method inaccuracy can be evaluated by comparing the total costs reduction obtained using forecasts (70%) and the one related to the optimal solution (82.3%).

REFERENCES

Dixon, P.S., Silver, R.A., 1981. A heuristic solution procedure for the multi-item single level, limited capacity, lot sizing problem. *Journal of Operations Management*, 2, 23-29.

Drexl, A., Kimms, A., 1997. Lot sizing and scheduling – survey and extensions. *European Journal of Operational Research*, 100, 494-516.

Karimi, B., Fatemi Ghomi, S.M.T., Wilson, J.M, 2003. The capacitated lot-sizing: a review of models and algorithms. *Omega*, 31, 365-378

Lee, T.S., Adam, E.E., 1986. Forecasting error evaluation in material requirements planning (MRP) production-inventory systems. *Management Science*, 32, 1186-1205.

Silver, E.A., Meal, H.C., 1973. A heuristic for selecting lot size quantities for the case of a deterministic time-varying demand rate and discrete opportunities for replenishment. *Production and Inventory Management Journal*, 2nd Quarter, 64-74.

Wagner, H., Whitin, T.M., 1958. Dynamic version of the economic lot size model. *Management Science*, 5(1), 89-96.

Xie, J., Lee, T.S., Zhao, X., 2004. Impact of forecasting error on the performance of capacitated multi-item production systems. *Computers & Industrial Engineering*, 46, 205-219.

Zhao, X., Goodale, J., Lee, T.S., 1995. Lot-sizing rules and freezing the master production schedule in MRP systems under demand uncertainty. *International Journal of Production Research*, 33, 2241-2276.

AUTHORS BIOGRAPHY

Lorenzo Tiacci, Laurea degree in Mechanical Engineering, Doctoral degree in Industrial Engineering, is post-doctoral fellow at the Department of Industrial Engineering of the University of Perugia (Italy). His research activity is focused on the use of discrete-event simulation, genetic algorithms, and other operations research techniques to aid in optimisation and decision making about industrial engineering problems, such as: production system design, planning and control; inventory control; supply chain management.

Stefano Saetta, Laurea degree in Electrical Engineering, is Associate Professor of Industrial Plants at the Faculty of Engineering of the University of Perugia (Italy). His research activity is focused on simulation and optimization of the supply chain, industrial plant simulation, decision supporting tools and operations research.

STUDY OF VEHICULAR INTERACTION IN HETEROGENEOUS TRAFFIC FLOW ON INTERCITY HIGHWAYS USING MICROSCOPIC SIMULATION

V. Thamizh Arasan [(a)] & Shriniwas S. Arkatkar [(b)]

[(a)] Professor, Transportation Engineering Division, Department of Civil Engineering,
Indian Institute of Technology Madras, Chennai - 600 036, India
[(b)] Ph.D. scholar, Transportation Engineering Division, Department of Civil Engineering,
Indian Institute of Technology Madras, Chennai - 600 036, India

[(a)] arasan@iitm.ac.in, [(b)] s_arkatkar@yahoo.co.in

ABSTRACT

Study of the basic traffic flow characteristics and comprehensive understanding of vehicular interaction are the pre-requisites for highway capacity and level of service analyses and formulation of effective traffic regulation and control measures. This is better done by modeling the system, which will enable the study of the influencing factors over a wide range. Computer simulation has emerged as an effective technique for modelling traffic flow due to its capability to account for the randomness related to traffic. This paper is concerned with application of a simulation model of heterogeneous traffic flow, named HETEROSIM, to study the relationships between traffic flow variables such as traffic volume and speed. Further, the model is also applied to quantify the vehicular interaction in terms of Passenger Car Equivalent (PCE) or Passenger Car Unit (PCU), taking a stretch of an intercity road in India as the case for the study. The results of the study, provides an insight into the complexity of the vehicular interaction in heterogeneous traffic.

Keywords: Heterogeneous Traffic, Micro-Simulation, Passenger Car Unit and Highway Capacity.

1. INTRODUCTION

The road traffic in the developing countries like India is highly heterogeneous comprising vehicles of wide ranging static and dynamic characteristics. The different types of vehicles present in the traffic can be broadly grouped into eight different categories as follows: 1. Motorized two-wheelers, which include motor cycles, scooters and mopeds, 2. Motorized three-wheelers, which include Auto-rickshaws – three wheeled motorized transit vehicles to carry a maximum of three passengers and tempos – three wheeled motorized vehicles to carry small quantities of goods, 3.Cars including jeeps and small vans, 4. Light commercial vehicles comprising large passenger vans and small four wheeled goods vehicles, 5. Buses, 6. Trucks, 7. Bicycles and 8.Tricycles, which include cycle-rickshaws- three wheeled pedal type transit vehicles to carry a maximum of two passengers and three wheeled pedal type vehicles to carry small amount of goods over short distance. These motorised and non-motorised vehicles share the same road space without any physical segregation. The speeds of these vehicles vary from just 5 to over 100 km/h. Due to the highly varying physical dimensions and speeds; it becomes difficult to make the vehicles to follow traffic lanes. For manoeuvre, the vehicles take any lateral position along the width of roadway, based on space availability. When such different types of vehicles having varying static and dynamic characteristics mix and move on the same roadway facility, a variable set of longitudinal and transverse distribution of vehicles are noticed from time to time.

The study of vehicular interaction is intended to quantify the relative impact of the presence of each of the different types of vehicles on traffic flow. This can be achieved by estimating Passenger Car Unit (PCU) values for the different categories of vehicle in the traffic. Under heterogeneous traffic conditions, in India, expressing traffic volume as number of vehicles per hour per lane is irrelevant and the volume of traffic has to be expressed taking the whole of the width of roadway as the basis. Also, the volume of such heterogeneous traffic needs to be expressed as PCU per hour by converting the different types of vehicles into equivalent passenger cars. Hence, estimation of PCU values of different categories of vehicles at various traffic volume levels is necessary for planning, design, and operational analysis of roadway facilities, in addition to regulation and control of traffic.

To arrive at an estimate of the PCU values, it is necessary to study the influence of roadway and traffic characteristics and the other relevant aspects, on vehicular movement, accurately. Study of these by observing various aspects of traffic flow in the field is difficult and time consuming. Also, it is not possible to carry out such experiments in the field covering a wide range of traffic volume and composition on a given roadway due to practical difficulties. Hence, it is necessary to model road–traffic flow for in depth understanding of the related aspects. The study of these complex characteristics, that may not be sufficiently simplified using analytical solution, requires alternative tools like computer simulation (Banks et al. 2004). Simulation, from microscopic through macroscopic, is increasingly becoming a popular traffic-flow modeling tool for analyzing traffic operations and highway capacity. Helbing et al. (2002), have shown that all the presently known macroscopic phenomena of freeway traffic, including (i) the fundamental diagrams, (ii) the characteristic parameters of congested traffic and (iii) the transitions between free traffic and other congested traffic states can be reproduced and explained by microscopic and macroscopic traffic models based on plausible assumptions and realistic parameters.

This paper is focused on the conceptual traffic simulation framework of highly heterogeneous traffic flow and application of the microscopic simulation model to study the relationship between traffic volume and speed. The model is also applied to study vehicular interaction by quantifying the relative impact of the presence of each of the different types of vehicles on traffic flow, under homogeneous (cars-only) and heterogeneous traffic conditions, at various traffic volume levels taking all the influencing factors into account.

2. OBJECTIVE AND SCOPE OF THE STUDY

Most traffic and transportation system simulation applications today are based on the simulation of vehicle-vehicle interactions and are microscopic in nature. The interaction between moving vehicles under heterogeneous traffic condition is highly complex. Microscopic simulation is a very powerful technique and has been applied to study the complex nature of the vehicular interactions in traffic stream. The knowledge of traffic volume is an important basic input required for planning, analysis and operation of roadway systems. Expressing traffic volume as number of vehicles passing a given section of road or traffic lane per unit time will be inappropriate when several types of vehicles with widely varying static and dynamic characteristics are comprised in the traffic. The problem of measuring volume of such heterogeneous traffic has been addressed by converting the different types of vehicles into equivalent passenger cars and expressing the volume in terms of Passenger Car Unit (PCU) per hour. Hence, the objective of the research work reported here is to quantify the vehicular interaction, in terms of Passenger Car Unit (PCU) values, of different categories of vehicles at various traffic volume levels, under heterogeneous traffic conditions prevailing on intercity roads, in plain terrain, in India. A recently developed heterogeneous traffic-flow simulation model, named, HETEROSIM is used to study the vehicular interactions, at micro-level, over a wide range of traffic flow conditions. Field data collected on traffic flow characteristics such as free speed, acceleration, lateral clearance between vehicles, etc. are used for validation of the simulation model. The validated model is then applied to develop the relationship between traffic volume and speed and derive Passenger Car Unit (PCU) values for different types of vehicles. Finally, check for the accuracy of the estimated PCU values is also made.

3. THE SIMUALTION FRAMEWORK

Simulation models may be classified as being static or dynamic, deterministic or stochastic, and discrete or continuous. A simulation model, which does not require any random values as input, is generally called *deterministic*, whereas a *stochastic* simulation model has one or more random variables as inputs. Random inputs lead to random outputs and these can only be considered as estimates of the true characteristics of the system being modeled. Discrete and continuous models are defined in an analogous manner. The choice of whether to use a discrete or continuous simulation model is a function of the characteristics of the system and the objectives of the study (Banks et al. 2004). For this study, a dynamic stochastic type discrete event simulation is adopted in which the aspects of interest are analysed numerically with the aid of a computer program.

The applications of traffic simulation programs can be classified in several ways. According to the problem area one can separate intersection, mid-block road section and network simulations. For traffic and transportation system applications, the available traffic-simulation-program packages have been used by the researchers all over the world. Bloomberg and Dale (2000) have given the detailed information about the use of two popular traffic simulation models (CORSIM and VISSIM) for traffic analysis on a congested network. Ben-Akiva et al. (1997), developed a simulation laboratory for performance evaluation and design refinement of dynamic traffic management systems. The simulation laboratory has been implemented in C++ using object-oriented programming and a distributed environment. Elefteriadou et al. (1997), used simulation as a tool to develop a methodology for calculating passenger car equivalents for freeways, two-lane highways, and arterials. Ahn et al. (2002), estimated vehicle fuel consumption and emissions based on instantaneous speed and acceleration using INTEGRATION microscopic simulation model. AIMSUN, DRACULA, PARAMICS and VISSIM are the main micro-simulation tools that have been used to model traffic on UK roads (Barcelo 1996).

As this research work pertains to the heterogeneous traffic conditions prevailing in India, the available traffic-simulation-program packages mentioned above such as CORSIM, AIMSUN, VISSIM, etc. cannot be directly used to study the characteristics of the traffic flow as these are based on homogeneous traffic-flow conditions. Also, the research attempts made earlier (Khan and Maini 2000; Marwah and Singh 2000; Kumar and Rao 1996; and Ramanayya 1988) to simulate heterogeneous traffic flow on Indian roads were limited in scope as they were location and traffic-condition specific. Moreover, these studies did not truly represent the absence of lane and queue discipline in heterogeneous traffic. Hence, an appropriate traffic simulation model, named, HETEROSIM has been developed (Arasan and Koshy 2005) to replicate heterogeneous traffic flow conditions accurately.

The modelling framework is explained briefly here to provide the background for the study. For the purpose of simulation, the entire road space is considered as single unit and the vehicles are represented as rectangular blocks on the road space, the length and breadth of the blocks representing respectively, the overall length and the overall breadth of the vehicles. The entire road space is considered to be a surface made of small imaginary squares (cells of convenient size 100 mm in this case); thus, transforming the entire space into a matrix. The vehicles will occupy a specified number of cells whose co-ordinates would be defined before hand. The front left corner of the rectangular block is taken as the reference point, and the position of vehicles on the road space is identified based on the coordinates of the reference point with respect to an origin chosen at a convenient location on the space. This technique will facilitate identification of the type and location of vehicles on the road stretch at any instant of time during the simulation process (Fig. 1).

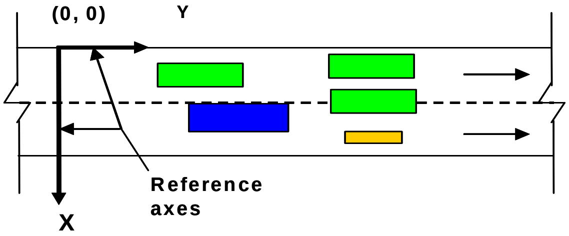


Figure 1: Reference Axes for Representing Vehicle Positions

The simulation model uses the interval scanning technique with fixed increment of time. For the purpose of simulation, the length of road stretch as well as the road width can be varied as per user specification. The model was implemented in *C*++ programming language with modular software design. The flow diagram illustrating the basic logical aspects involved in the program is shown as Figure 2. The simulation process consists of the following major sequential steps: (1) vehicle generation, (2) vehicle placement, and (3) vehicle movement.

3.1. Vehicle Generation

In a stochastic traffic simulation process, the vehicles arrive randomly, and they may have varying characteristics (e.g. speed and vehicle type). Traffic-simulation models therefore, require randomness to be incorporated to take care of the stochasticity. This is easily done by generating a sequence of random numbers. For generation of headways, free speed, etc., of vehicles, the model uses several random number streams, which are generated by specifying separate seed values. Whenever a vehicle is generated, the associated headway is added to the sum of all the previous headways generated to obtain the cumulative headway. The arrival of a generated vehicle occurs at the start of the warm-up road stretch when the cumulative headway equals the simulation clock time. At this point of time, after updating the positions of all the vehicles on the road stretch, the vehicle-placement logic is invoked.

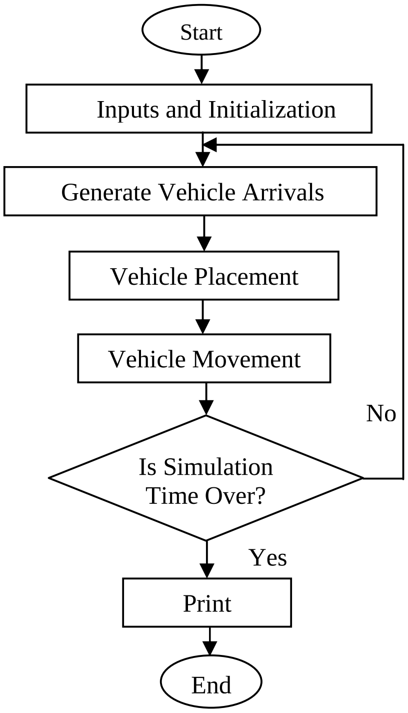


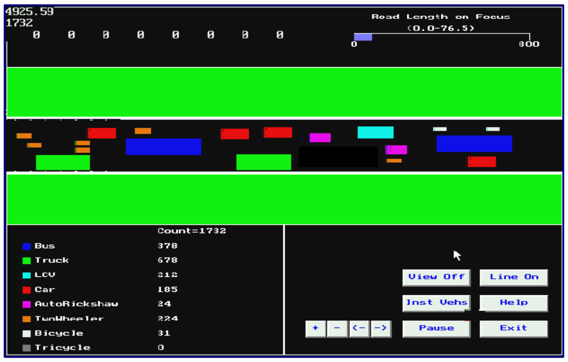
Figure 2: Flow Diagram of the Simulation Model

3.2. Vehicle Placement

Any generated vehicle is placed at the beginning of the simulation stretch, considering the safe headway (which is based on the free speed assigned to the entering vehicle), lateral gap and the overall width of the vehicle with lateral clearances. If the longitudinal gap in front is less than the minimum required safe gap, the entering vehicle is assigned the speed of the leading vehicle, and once again the check for safe gap is made. If the gap is still insufficient to match the reduced speed of the entering vehicle, it is kept as backlog, and its entry is shifted to the next scan interval. During every scan interval, the vehicles remaining in the backlog will be admitted first, before allowing the entry of a newly generated vehicle.

3.3. Vehicle Movement

This module of the program deals with updating the positions of all the vehicles in the study road stretch sequentially, beginning with the exit end, using the formulated movement logic. Each vehicle is assumed to accelerate to its free speed or to the speed limit specified for the road stretch, whichever is minimum, if there is no slow vehicle immediately ahead. If there is a slow vehicle in front, the possibility for overtaking the slow vehicle is explored. During this phase, the free longitudinal and transverse spacing available for the subject vehicle (fast moving vehicle), on the right and left sides of the vehicle in front (slow vehicle), are calculated. If the spacing is found to be adequate (at least equal to the movable distance of the vehicle intending to overtake plus the corresponding minimum spacing in the longitudinal direction and the minimum required lateral spacing in the transverse direction), an overtaking maneuver is performed. If overtaking is not possible, the fast vehicle decelerates to the speed of the slow vehicle in front and follows it. Thus, the various maneuvers for a vehicle moving on the simulation road stretch include free forward movement with desired speed, acceleration maneuver, movements leading to lateral shifting and overtaking of slower vehicles, movements involving deceleration and following of the front vehicle for want of sufficient gaps for overtaking, etc. The model is also capable of displaying the animation of simulated traffic flow through mid block sections. The animation module of the simulation model displays the model's operational behavior graphically during the simulation runs. The snapshot of animation of heterogeneous traffic flow, obtained using the animation module of HETEROSIM, is shown in Figure 3. The model has been applied for a wide range of traffic conditions (free flow to congested flow conditions) and has been found to replicate the field observed traffic flow to a satisfactory extent through an earlier study (Arasan and Koshy, 2005).

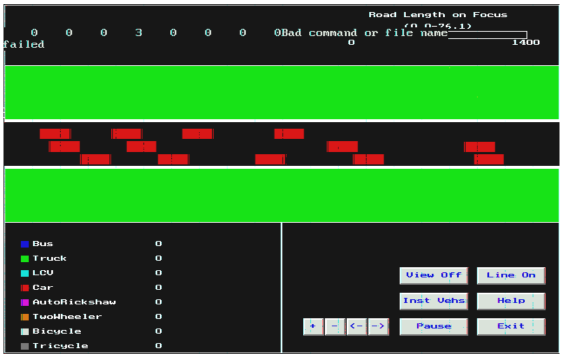


Count=1732

🟦 Bus	378
🟩 Truck	678
🟦 LCV	212
🟥 Car	185
🟪 AutoRickshaw	24
🟧 TwoWheeler	224
⬜ Bicycle	31
⬛ Tricycle	0

Figure 3: Snapshot of Animation of Simulated Heterogeneous Traffic Flow

For the purpose of simulation, the time scan procedure is adopted. The scan interval chosen for the simulation is 0.5 second. The arrival of vehicles on the road stretch will be checked for every 0.5 second and the arrived vehicles will be put on to the entry point of the study stretch of the road, on first-come-first-served basis. In the vehicle-generation module, the first vehicle is generated after initialization of the various parameters required to simulate heterogeneous traffic flow. Then, the generated vehicle is added to the system when the current time (clock time) becomes equal to the cumulative headway. At this stage, the module for adding vehicles named 'Add Vehicle' will be activated to facilitate the process. At higher traffic flow levels, there is a chance of more than one vehicle arriving during each scan interval (0.5s). To address this issue, an additional clock for scanning with a precision of 0.05 s is provided, so that a maximum of 20 vehicles can be added in one second. The precision of 0.05 s, decided based on field studies, is intended to account for the maximum possible number of smaller vehicles, like motorised two wheelers, auto-rickshaw, etc. that may arrive in large numbers in short periods on multilane highways. Thus, the logic formulated for the model also permit admission of vehicles in parallel across the road width, since it is common for smaller vehicles such as Motorised two-wheelers to move in parallel in the traffic stream without lane discipline. Vehicles admitted to the simulation road stretch are then allowed to move based on the various movement logics formulated. When the cumulative precision time is equal to the scan interval, the module for vehicle movement 'Move All Vehicles' will be activated to move all the vehicles in the simulation road stretch, with their current parameter values. The above process will be continued until the clock time matches with the assigned total simulation time. The model is also capable of simulating homogeneous traffic (cars-only traffic stream, comprising of 100 percentage of car). The snapshot of animation of homogeneous traffic flow, obtained using the animation module of HETEROSIM, is shown in Figure 4.



🟦 Bus	0
🟩 Truck	0
🟦 LCV	0
🟥 Car	0
🟪 AutoRickshaw	0
🟧 TwoWheeler	0
⬜ Bicycle	0
⬛ Tricycle	0

Figure 4: Snapshot of Animation of Simulated Homogeneous (Cars-only) Traffic Flow

4. DATA COLLECTION

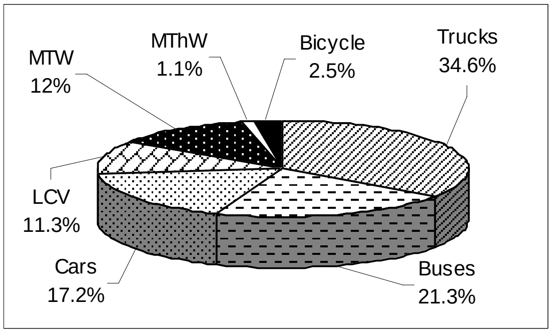
4.1. Study Stretch

The stretch of intercity roadway between km 77.2 and km 77.4, of National Highway No. 45 between the cities, Chennai and Chengalpet, in the southern part of India, was selected for collection of traffic data for the study. The study stretch is a four-lane divided road with 7.5 m wide main carriageway and 1.25 m of paved shoulder for each direction of movement. The stretch is straight and level with no side road connections. Also, the traffic flow on the study stretch was unhindered by the road side land uses.

4.2. Traffic Characteristics

Collection and analysis of data play a pivotal role in the development of successful simulation models. The field data inputs required for the model were collected at the selected stretch, which had a total carriageway width (including shoulder) of 8.75 m for each direction. A digital video camera was used to capture the traffic flow for a total duration of 1h. The video captured traffic data was then transferred to a Work station (computer) for detailed analysis.

The inputs required for the model to simulate the heterogeneous traffic flow are: road geometry, traffic volume, and composition, vehicle dimensions, minimum and maximum lateral spacing between vehicles, minimum longitudinal spacing between vehicles, free speeds of different types of vehicles, acceleration and deceleration characteristics of vehicles, the type of headway distribution and the simulation period. The required input traffic data for the simulation was obtained by running the video of the traffic flow at a slower speed (⅛th of the actual speed) to enable one person to record the data by observing the details displayed on the monitor of the computer. The composition of the measured traffic volume on the study stretch is as depicted in Figure 5. It may be noted that Animal drawn vehicles and Tricycles, which may be present in small numbers on certain intercity roads, are not present on the study stretch.



L.C.V. - Light Commercial Vehicles, M.Th.W. – Motorised Three-Wheelers, M.T.W. - Motorised Two-Wheelers

Figure 5: Traffic Composition at the Study Road Stretch

The free speeds of the different categories of vehicles were also estimated by video capturing the traffic under free-flow conditions. The speeds of the different categories of vehicles were measured by noting the time taken by the vehicles to traverse a trap length of 30 m. The observed mean, minimum and maximum free speeds of various classes of vehicles and their corresponding standard deviations are shown in columns (2), (3) ,(4) and (5) respectively of Table 1. The overall dimensions of all categories of vehicles, adopted from literature (Arasan and Koshy 2005), are shown in columns (2) and (3) of Table 2. Any vehicle moving in a traffic stream has to maintain sufficient lateral clearance on the left and right sides with respect to other vehicles/curb/ median to avoid side friction. These lateral clearances depend upon the speed of the vehicle being considered, speed of the adjacent vehicle in the transverse direction, and their respective types.

Table 1: Free Speed Parameters of Different Types of Vehicles

Vehicle type (1)	Free speed parameters in km/h			
	Mean (2)	Min. (3)	Max. (4)	Std. Deviation (5)
Trucks	62	53	90	8
Buses	70	45	90	10
Cars	86	60	110	15
L.C.V.	67	50	90	6
M.T.W	57	35	75	11
M.Th.W	52	45	55	3
Bicycles	14	10	20	4.5

L.C.V. - Light Commercial Vehicles, M.Th.W. – Motorised Three-Wheelers, M.T.W. - Motorised Two-Wheelers

Table 2: Observed Vehicle Dimensions

Vehicle type (1)	Average overall dimension (m)	
	Length (2)	Width (3)
Trucks	7.5	2.5
Buses	10.3	2.5
Cars	4.0	1.6
L.C.V.	5.0	2.0
M.T.W	2.0	0.75
M.Th.W	3.0	1.5
Bicycles	1.9	0.5

L.C.V. - Light Commercial Vehicles, M.Th.W. – Motorised Three-Wheelers, M.T.W. - Motorised Two-Wheelers

The minimum and maximum values of lateral-clearance share adopted from an earlier study (Arasan and Koshy 2005), are given in columns (2) and (3), respectively, of Table 3. The minimum and the maximum clearance- share values correspond to, respectively, zero speed and free speed conditions of respective vehicles. The lateral-clearance-share values are used to calculate the actual lateral clearance between vehicles based on the type of the subject vehicle and the vehicle by the side of it. For example, at zero speed, if a motorized two-wheeler is beside a car, then, the clearance between the two vehicles will be 0.2 + 0.3 = 0.5m. The data on, acceleration values of different vehicle categories, at various speed ranges, taken from available literature (Arasan and Koshy 2005), are shown in Table 4.

The observed traffic volume and composition was given as input to the simulation process. The simulation runs were made with different random number seeds and the averages of the values were taken as the final model output. The model output includes the number of each category of vehicle generated, values of all the associated headways generated, number of vehicles present over a given road length at any point of time, number of overtaking maneuvers made by each vehicle, speed profile of vehicles, etc.

Table 3: Minimum and Maximum Lateral Clearances

Vehicle type (1)	Lateral-clearance share (m)	
	At zero speed (2)	At a speed of 60 km/h (3)
Trucks	0.3	0.6
Buses	0.3	0.6
Cars	0.3	0.5
L.C.V.	0.3	0.5
M.T.W	0.1	0.3
M.Th.W	0.2	0.4
Bicycles	0.1	0.3*

L.C.V. - Light Commercial Vehicles, M.Th.W. – Motorised Three-Wheelers, M.T.W. - Motorised Two-Wheelers

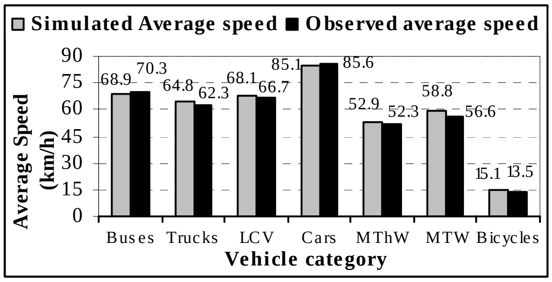
Table 4: Acceleration Rates of Different Categories of Vehicles

Vehicle type (1)	Rate of acceleration at various speed ranges (m/s^2)		
	0-20 km/h (2)	20- 40 km/h (3)	Above 40 km/h (4)
Trucks	0.80	0.6	0.50
Buses	0.90	0.75	0.60
Cars	1.40	1.10	0.95
L.C.V.	1.00	0.55	0.45
MTW	1.40	0.80	0.65
MThW	1.00	0.55	0.45
Bicycle	0.10	-	-

*- Maximum speed of these vehicles is 20 km/h

5. MODEL VALIDATION

A developed model needs to be validated to check whether the model is capable of replicating the real life situations (Field conditions) accurately or not. The most definitive test of a simulation model's operational validity is establishing that its output data closely resemble the output data that would be expected from the actual system using identical inputs (Law and Kelton 1991). For the purpose of validation, the simulation model was used to replicate the heterogeneous traffic flow on a stretch of road. The total length of road stretch, for simulation purpose, was taken as 1,400 m. The initial 200 m length, at the entry point, was used as a warm-up zone. To avoid unstable traffic flow condition at the exit end, a 200 m long road stretch at the exit end was also excluded from the analysis. Thus, the middle 1000 m length of the simulation stretch was used to collect the data of the simulated traffic flow characteristics. To eliminate the initial transient nature of traffic flow, the simulation clock was set to start only after the first 50 vehicles reached the exit end of the road stretch. The simulation model was run with three random number seeds, and the average of the three runs was taken as the final output of the model. The observed roadway condition, traffic volume and composition were given as input to the simulation process. The inter arrival time (headway) of vehicles was found to fit into negative exponential distribution and the free speeds of different categories of vehicles, based on the results of an earlier study (Arasan and Koshy 2005), was assumed to follow Normal distribution. These distributions, then, formed the basis for input of the two parameters for the purpose of simulation. To check for the validity of the model, the vehicle speeds simulated by the model were compared with the field observed speed values for each category of vehicles. The results of the experiment, for the observed traffic volume of 482 vehicles per hour, are shown in figure 6. It can be seen that the simulated speed values significantly replicate the field observed speeds for all vehicle types.



L.C.V. - Light Commercial Vehicles, M.Th.W. – Motorised Three-Wheelers, M.T.W. - Motorised Two-Wheelers

Figure 6: Model Validation by Comparison of Speeds

A statistical validation of the model, based on observed and simulated speeds of different categories of vehicles, was also done through t-test. The value of t-statistic, calculated based on the observed data (t_0), is 1.39. The critical value of t statistic for level of significance of 0.05 (95% confidence limit), at 6 degrees of freedom, obtained from standard t-distribution table is 2.45. Thus, it can be seen that the value of t statistic, calculated based on the observed data, is less than the corresponding table value. This implies that there is no significant difference between the simulated and observed means speeds.

6. MODEL APPLICATION

The 'HETEROSIM' model can be applied to study various heterogeneous traffic scenarios for varying traffic and roadway conditions. Here, the application of the model is specific to develop relationship between traffic volume and speed and then to quantify the relative impact of the presence of each of the different types of vehicles on traffic flow by estimating PCU value under heterogeneous traffic conditions.

6.1. Speed-Volume Relationship

One of the basic studies in traffic flow research is pertaining to the relationship between speed and volume of traffic. The highway capacity for different roadway and traffic conditions can be estimated using speed-volume relationship. Hence, the speed-flow relationship was developed for the heterogeneous traffic flow, the composition of traffic and roadway conditions being the same as observed in the field, by running the simulation for various volumes, starting from near zero to the capacity of the road. Also, speed-volume relationship for cars-only traffic (traffic stream comprising of 100 percentage cars) was developed by simulating the homogeneous traffic flow from the minimum to the maximum possible volumes.

The total length of road stretch considered for the experiments is 1400 m, with 200 m sections at the entry and exit excluded from output data collection as warm-up and stabilizing section. The central 1000 m stretch was considered as the observation stretch, the various traffic flow parameters were recorded while vehicles were moving through it. To account for the variation due to randomness, the simulation runs were repeated using three different-random number streams to check for the consistency of the results. Both the speed-volume relationships pertaining to 8.75 m wide road are depicted, on the same set of axes, in figure 7. It can be seen that, in both the cases, the speed-volume curves follow the established trend. Also, it can be seen from the speed-volume curves, the capacity of the considered road stretch is about 2700 vehicles per hour under the heterogeneous traffic condition and it is about 4500 cars per hour under cars-only traffic condition.

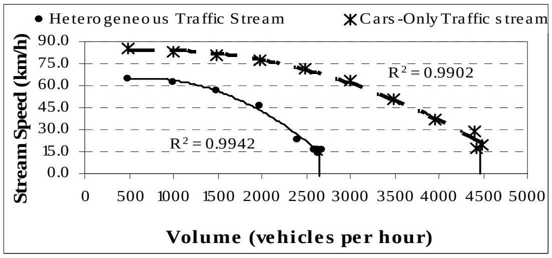


Figure 7: Speed – Volume Relationship

6.2. Estimation of PCU Values

Expressing highway-capacity (volume) as number of vehicles passing a given section of road per hour will be inappropriate when two or more than two types of vehicles with widely varying static and dynamic characteristics are present in the road traffic. The capacity-volume of such heterogeneous traffic can be expressed more precisely as Passenger Car Unit (PCU) per hour by converting the different types of vehicles into equivalent passenger cars. Therefore, it is very important to estimate these PCU values accurately. After a careful study of the various approaches adopted for estimation of PCU of vehicles, it was found that the methodology of approach of Transport and Road Research Laboratory (TRRL), London, UK may be appropriate for the heterogeneous traffic being dealt with. The PCU has been defined by TRRL (1965) as follows: *"on any particular section of road under particular traffic condition, if the addition of one vehicle of a particular type per hour will reduce the average speed of the remaining vehicles by the same amount as the addition of, say x cars of average size per hour, then one vehicle of this type is equivalent to x PCU.* This definition has been taken as the basis for derivation of PCU values, in this study. Hence, the PCU values for the different types of vehicles, at various volume levels, were estimated by taking the average stream speed as the measure of performance.

6.2.1. Estimating PCU Values in Cars-Only Traffic

Though the prime objective of this study is to quantify the vehicular interactions, in terms of Passenger Car Unit (PCU) under heterogeneous traffic, it will be appropriate to estimate the Passenger Car Unit (PCU), values of different vehicle types while moving with cars-only traffic stream to provide a set of basic PCU values of the different types of vehicles for the purpose of comparison. This will provide information on the absolute amount of impedance caused by a vehicle type while moving in the traffic stream, which comprises of cars and the subject vehicles only.

Since, speed is the performance measure identified to estimate the PCU values, average speed of cars-only traffic for a set of selected volume levels corresponding to volume-to-capacity ratios of 0.13, 0.25, 0.38, 0.50, 0.63, 0.75, 0.88 and 1.0 (taking the capacity value from the speed-flow curve corresponding to cars only traffic shown in figure 7) were estimated by simulating the homogeneous traffic flow (100 % passenger cars) in one direction, on four-lane, divided intercity road. The impedance caused by a vehicle type, in terms of PCU, for a chosen volume level was estimated by replacing a certain percentage (the observed percentage composition of the subject-vehicle in the field - Fig. 5) of cars in the homogeneous traffic stream with the subject-vehicle type, such that, the average speed of cars remained the same as before the replacement of the cars. The number of subject vehicle can be adjusted on trial basis by observing the average speed of cars in each trial. If the average car speed is more, after replacement, than the average car speed under homogeneous traffic, it is to be inferred that, the introduced number of subject vehicles is inadequate to compensate for the removed cars. Similarly, if the average speed of cars, after replacement, is less than the average car speed under homogeneous traffic, it is to be inferred that the introduced subject-vehicle volume is more than the equivalent volume of cars. After regaining the original speed of cars by adjusting the number of subject vehicles, the PCU value of the vehicle type can be estimated using the relation,

$$PCU\ Value\ of\ subject-vehicle\ type=\frac{Number\ of\ cars\ removed}{Number\ of\ subject-vehicle\ type\ added} \tag{1}$$

The logic behind the above approach is that, as stated in the definition of PCU, the introduced subject vehicle type creates more or less the same effect on the traffic stream that is equivalent to that of the cars removed from the stream. The PCU value of the subject-vehicle was determined, following the said procedure, for the same set of traffic volume levels selected for cars-only traffic. To account for the variation due to randomness, the simulation runs were made with three random number seeds and the average of the three values was taken as the final value. The variation of PCU values of the different types of vehicles over traffic volume, in homogeneous (Cars-only) traffic condition has been shown in Table 5.

Table 5: Variation of PCU Value over Volume for Different Vehicles Types in Cars-only Traffic

V/C ratio (1)	PCU value					
	Buses (2)	Trucks (3)	LCV (4)	MThW (5)	MTW (6)	Bicycle (7)
0.13	3.00	3.26	2.16	1.10	0.90	0.85
0.25	2.87	3.11	2.04	1.60	1.50	1.35
0.38	2.75	2.95	1.93	1.75	1.60	1.48
0.50	2.63	2.83	1.85	1.80	1.65	1.53
0.63	3.10	3.25	1.97	1.40	1.28	1.13
0.75	3.66	3.62	2.35	1.20	1.10	0.92
0.88	4.50	4.28	2.74	1.00	0.90	0.82
1.00	5.57	5.33	3.45	0.90	0.78	0.75

L.C.V. - Light Commercial Vehicles, M.Th.W. – Motorised Three-Wheelers, M.T.W. - Motorised Two-Wheelers

From table 5, it can be seen that at low volume levels, in the case of vehicles that are larger in size than car (columns (2), (3) and (4)), the PCU decreases with increase in traffic volume (when V/C ratio is less than 0.63) and the PCU increases with the increase in traffic volume at high volume levels (When V/C ratio is 0.63 and more). Whereas, in the case of vehicles that are smaller than car (columns (5), (6) and (7)), at low volume levels, the PCU increases with increase in traffic volume and the PCU decreases with increase in traffic volume at high volume levels. The attempt to find the possible reason for these trends revealed that the relative changes, caused by the overall traffic environment, (because of the factors such as manoeuvrability and physical size of the subject vehicle type) in the speeds of the reference vehicle (car) and the subject vehicle (for which the PCU value is to be estimated), at various traffic volume levels, are the main contributors to the trend.

6.2.2. Estimating PCU Values in Heterogeneous Traffic

The PCU values for the different types of vehicles, at various volume levels, were estimated using simulation. For the purpose of simulation, eight traffic volume levels corresponding to volume to capacity (V/C) ratios of 0.13, 0.25, 0.38, 0.50, 0.63, 0.75, 0.88 and 1.0 (taking the capacity value from the speed-flow curve corresponding to heterogeneous traffic shown in figure 7) were considered. At each volume level, first, heterogeneous traffic flow of field observed composition (figure 5) was simulated for an hour and the traffic stream speed was obtained as the weighted average of the speeds of the different categories of vehicles. Then, a certain percentage of cars were replaced by the subject vehicle type (for which the PCU value is to be estimated) in the mixed traffic stream, such that the average stream speed obtained by simulation (figure 7), remained the same as the earlier stream speed. Then, for each flow level, the number of cars removed divided by the number of subject vehicle type introduced will give the PCU value of that vehicle type. The variation of PCU values of the different types of vehicles over traffic volume, in heterogeneous traffic condition, for the purpose of comparison, has been presented in Table 6. It can be seen that the general trend of variation of the PCU values of vehicles over volume is the same as in the case of cars-only traffic. Hence, the explanation provided for the trend in the case of cars-only traffic is valid for heterogeneous traffic condition also.

Table 6: Variation of PCU Value over Volume for Different Vehicles Types in Heterogeneous Traffic

V/C ratio (1)	PCU value					
	Buses (2)	Trucks (3)	LCV (4)	MThW (5)	MTW (6)	Bicycle (7)
0.13	2.00	2.25	1.42	0.50	0.34	0.30
0.25	1.95	2.20	1.38	0.72	0.43	0.42
0.38	1.90	2.15	1.32	0.85	0.52	0.54
0.50	1.80	2.10	1.28	0.90	0.66	0.66
0.63	1.70	1.90	1.24	0.85	0.74	0.72
0.75	1.80	1.95	1.28	0.80	0.72	0.70
0.88	2.20	2.10	1.32	0.72	0.62	0.63
1.00	2.70	2.50	1.48	0.60	0.49	0.50

L.C.V. - Light Commercial Vehicles, M.Th.W. – Motorised Three-Wheelers, M.T.W. - Motorised Two-Wheelers

6.3. Effect of Heterogeneity on PCU values

It is clear that the degree of heterogeneity of traffic stream affects the speed and other traffic flow parameters, and influences the magnitude of interaction between the moving vehicles significantly. The presence of a vehicle type, other than car, in the cars-only traffic stream, creates a traffic condition, which is totally different from the cars-only traffic condition. The change in the traffic condition make the vehicles to offer varying amount of impedance to the movement of adjacent vehicles in the traffic stream, depending upon the extent of variation of traffic stream from cars-only (homogeneous) traffic condition. In the light of the said fact, a comparison of the interactions of different vehicle types in cars-only traffic and in heterogeneous traffic, the amount of interaction having been measured in terms of PCU, will be useful. Figures 8 through 13 illustrate the comparison of PCU values of different vehicle type and their variations over traffic volume, in cars-only traffic and heterogeneous traffic flow conditions. It may be noted that, to facilitate plotting of the variation of PCU in homogeneous and heterogeneous traffic conditions using the same set of axes, the traffic volume has been represented using V/C ratio.

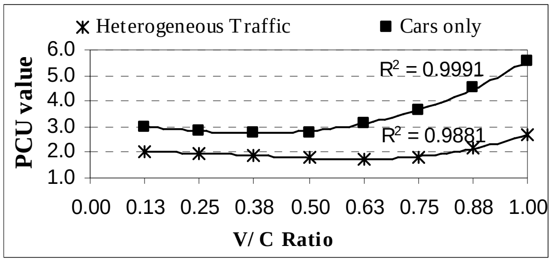


Figure 8: Variation of PCU Values of Buses on 8.75 m Wide Road

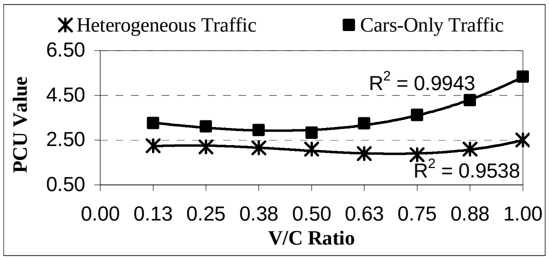


Figure 9: Variation of PCU Values of Trucks on 8.75 m Wide Road

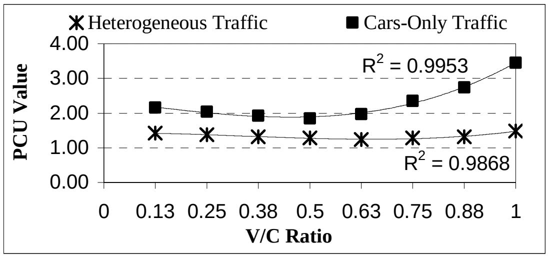


Figure 10: Variation of PCU Values of Light Commercial Vehicles on 8.75 m Wide Road

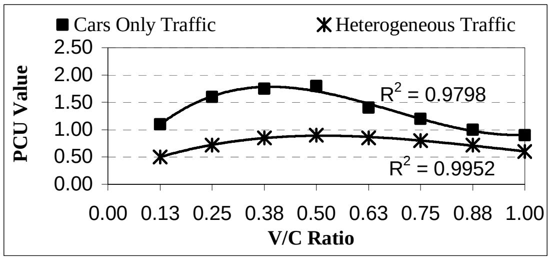


Figure 11: Variation of PCU Values of Motorised Three-Wheelers on 8.75 m Wide Road

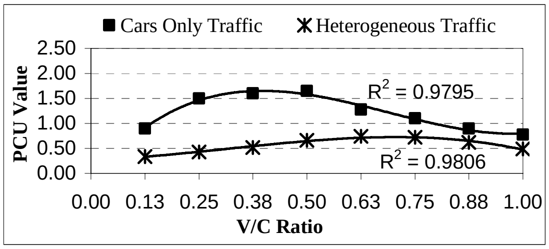


Figure 12: Variation of PCU Values of Motorised Two-Wheelers on 8.75 m Wide Road

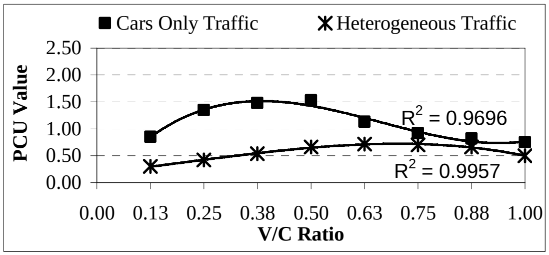


Figure 13: Variation of PCU Values of Bicycles on 8.75 m Wide Road

It can be seen that, the magnitude of vehicular interactions measured in terms of Passenger Car Units (PCU), under cars-only traffic condition, are significantly higher for all the vehicle types, when compared to their corresponding values under heterogeneous traffic condition. Higher PCU values under cars-only traffic condition may be attributed to the higher speed difference between the cars and the subject-vehicle speed in cars-only traffic than the difference between car speed and subject-vehicle speed under heterogeneous traffic condition. For example, at volume-to-capacity ratio value of 0.63, under cars-only traffic condition, through the simulation experiments it has been found that the average speed of cars is 74.19 km/h and buses is 58.76 km/h, with a speed difference of 15.43 km/h. Whereas under heterogeneous traffic condition, the average car speed for the same volume-to-capacity ratio is 58.01 km/h and the average bus speed is 51.23 km/h, resulting in a speed difference of 6.78 km/h. The PCU values of buses at this level of traffic flow under cars-only traffic and heterogeneous traffic conditions are 3.1 and 1.7 respectively.

7. CHECK FOR ACCURACY OF PCU VALUES

For the purpose of checking the accuracy of the PCU estimates for the different categories of vehicles, first, the heterogeneous traffic flow of field observed composition was simulated for one hour period for selected values of V/C ratios and the number of vehicles in each category, for each case, was noted. Then, the vehicles of the different categories were converted into equivalent PCUs by multiplying the number of vehicles in each category, obtained for the selected V/C ratios, by the corresponding PCU values (Table:6). The products, thus, obtained were summed up to get the total traffic flow in PCU/h. Then, 'cars-only' traffic was simulated for one hour for the same set of V/C ratio values (taking the capacity value from the speed-flow curve corresponding to cars only traffic shown in Figure 7). Thus, the traffic volume, in terms of number of cars, was obtained for the set of selected V/C ratios. A comparison of the traffic flow in terms of PCU and in terms of number of passenger cars, for the set of the selected V/C ratios, is shown in Figure 14. It can be seen that the heterogeneous traffic flow in PCU/h and the cars-only flow in cars/h match to a greater extent at each V/ C ratio, indicating the accuracy of the estimated PCU values.

A paired t-test, based on the passenger cars equivalent (PCU/h) and passenger cars-only (cars/h) traffic volumes was also done. The value of t statistic calculated (t_0) is 0.82. The critical value of t statistic for a level of significance of 0.05 for 7 degrees of freedom, obtained from standard t-distribution table is 2.37. This implies that, there is no significant difference between the traffic volumes measured in terms of passenger cars and in PCU.

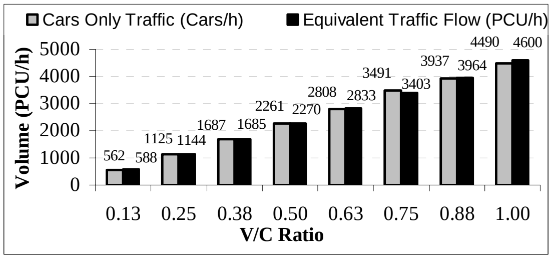


Figure 14: Comparison of Heterogeneous Traffic and Cars-only Traffic Flows

8. CONCLUSIONS

The following are the important conclusions of the study:

1. The simulation model of heterogeneous traffic flow named, HETEROSIM is found to be valid for simulating heterogeneous traffic flow on intercity roads to a satisfactory extent. The validity of the model is further confirmed by the speed-flow relationships developed, using the simulation model, which are found to follow the well established trend of the speed-volume curve.
2. From the speed-volume curve, developed using the simulation model, it is found that, for the observed traffic composition, the capacity of a four lane divided road with 7.5 m wide main carriageway and 1.25 m wide paved shoulder, for one direction of traffic flow, in plain terrain, is about 4600 PCU per hour.
3. It is found that, the estimated PCU values of the different categories of vehicles of the heterogeneous traffic are accurate at 5% level of significance.
4. It is found that, by virtue of the complex nature of interaction between vehicles under the heterogeneous traffic condition, the PCU estimates, made through simulation, for the different types of vehicles of heterogeneous traffic, for a wide range of traffic volume levels significantly changes with change in traffic volume.

REFERENCES

Ahn, K., Rakha, H., Trani, A.and Van Aerde, M. 2002. Estimating Vehicle Fuel Consumption and Emissions based on Instantaneous Speed and Acceleration. *ASCE Journal of Transp. Engg.*,128(2), 182-190.

Arasan, V.T. and. Koshy, R.Z., 2005. Methodology for Modeling Highly Heterogeneous Traffic Flow. *ASCE Journal of Transp. Engg.*, Vol. 131, No. 7, 544-551.

Banks, J.; Carson, J.S.; Barry, L.N.; and David, M.N., 2004. Discrete-Event System Simulation. Pearson Education, Singapore, Third Edition, 12-14.

Barcelo, 1996. The parallelization of AIMSUN2 Microscopic Traffic Simulator for ITS applications. Available from: http://www.microsimulation.drfox.org.uk /tools [accessed 5 June 2008].

Ben-Akiva, Moshe E., Koutsopoulos, H. N., Mishalani, R. G., and Yang Qi. 1997. Simulation Laboratory for Evaluating Dynamic Traffic Management Systems. *ASCE Journal of Transp. Engg.*,123(4), 283-289.

Bloomberg, L. and Dale, J. 2000. A comparison of the VISSIM and CORSIM Traffic Simulation Models on a Congested Network. *TRR: Journal of the TRB*, No. 1721, TRB, National Research Council, Washington, D. C., 52-60.

Elefteriadou, L., Torbic, D. and Webster, N. 1997. Development of Passenger Car Equivalents for Freeways, Two-Lane Highways, and Arterials.” *TR R: Journal of the TRB*, No. 1572, TRB, National Research Council, Washington, D. C., 51-58.

Helbing, D., Hennecke, A., Shvetsov, V, and Treiber M. 2002. Micro-and macro-simulation of freeway traffic. *Journal of Mathematical and Computer Modelling.* 35(5/6), 517-547.

Khan, S.I. and Maini, P., 2000. Modeling Heterogeneous Traffic Flow. *TRR*, 1678, 234-241.

Kumar, V.M. and Rao, S.K., 1996. Simulation Modeling of Traffic Operations on Two Lane Highways. *Highway Research Bulletin*, No. 54, Indian Roads Congress, 211-237.

Law, A. M., and Kelton, W. D. 1991. *Simulation Modelling and Analysis*. Mc-Graw-Hill Higher Education Singapore.

Marwah, B. R. and Singh, B., 2000. Level of Service Classification for Urban Heterogeneous Traffic: A Case Study of Kanpur Metropolis. *Transportation Research Circular E-C018: Proceedings of the 4th International Symposium on Highway Capacity*, Maui, Hawaii, 271–286.

Ramanayya, T. V., 1988. Highway capacity under mixed traffic conditions. *Traffic Engineering and Control*, U.K., 29(5), 284-287.

Transportation and Road Research Laboratory (TRRL) *Research on Road Traffic*, H.M.S.O, London, 1965.

AUTHORS BIOGRAPHIES

Prof. Dr. V. Thamizh Arasan is currently a full Professor in the Transportation Engineering Division of the Department of Civil Engineering of Indian Institute of Technology Madras, Chennai, India, which is one of the seven national level higher technological institutions in the country. He has a professional experience of about 30 years in teaching research and consultancy in the area of Transportation Engineering. Travel demand modeling and traffic flow modeling are his areas of research interest. He has guided a number of doctoral degree students and has published more than eighty research papers in international and national journals and conference proceedings. Three of his papers published in journals have received awards for excellence in research. Prof. Arasan has successfully completed several sponsored research projects both at national and international levels. The international projects are: (i) Development of Transportation Planning Techniques for Indian conditions in collaboration with the Technical University of Braunschweig, Germany and (ii) Enhancing the Level of Safety at Traffic Signals in collaboration with the Technical University of Darmstadt, Germany.

Mr. Shriniwas S. Arkatkar is a Ph.D. Scholar in Transportation Engineering Division, Department of Civil Engineering, Indian Institute of Technology Madras, Chennai, India. His doctoral research work is in the area of ‘Heterogeneous Traffic Flow Modeling’. Mr. Shriniwas S. Arkatkar obtained his undergraduate degree in the area of Civil Engineering in the year 1999 and post graduate degree in the area of Urban Planning in the year 2001 from Visvesvaraya National Institute of Technology, Nagpur, India. He has served as Lecturer in Department of Civil Engineering, Nirma University, Ahmedabad, India for the duration of three and half years.

GEOMETRIC AND MULTIBODY MODELING OF RIDER-MOTORCYCLE SYSTEM

M. Calì[(a)], S. M. Oliveri[(a)], G.Sequenzia[(a)], F. Trovato [(b)]

[(a)]*Università di Catania, Dipartimento di Ingegneria Industriale e Meccanica, viale A Doria, 6 – 95125 Catania*
[(b)]*Moto Guzzi S.p.A., via E. V. Parodi, 57 Mandello del Lario (Lecco)*

[(a)]mcali@diim.unict.it

ABSTRACT

In this study, a methodology based on co-simulation was developed for the multibody parametric modelling of a motorcycle with an anthropomorphic model of the rider. This co-simulation uses two different software programs, integrated to ensure a complete exchange of information between them in real time.

The paper reports the effects induced by the movement of the rider's body on the dynamics and performance of a motorcycle. The legs of an anthropomorphic model were used as kinematics to control transverse movements of the motorcycle.

The control system inputs are the geometric characteristics of the road (length, width and radius of curvature) and the speed of the vehicle along the track. For the dynamic behaviour of the motorcycle, the only channels currently operated by the control system are steering angle and engine torque, which are determined in accordance with the input parameters.

Keywords: motorcycle, rider, control, multibody, dynamic

1. INTRODUCTION

On the saddle of a motorcycle, the rider enjoys considerable mobility offering appreciable movement of his body and, thereby, of the barycentre of the rider/motorcycle system. The movements he makes are directed at improving the performance achieved by the motorcycle in the course of certain manoeuvres, so that, in the dynamics of a motorcycle, the rider's contribution is often a decisive factor in improving performance. It is therefore of interest to analyse the relation between the movements of the rider's body and the dynamic characteristics of the motorcycle.

Theoretical instruments have allowed the study of motorcycle dynamics only under certain conditions of motion, in the main constant, and often imposing rather restrictive hypotheses. The problem is commonly analysed using multibody software programs, without attention ever being focussed on how movements of the rider's body can influence the dynamic behaviour of the motorcycle, and certainly without ever implementing a system of active control which optimises these movements.

In previous research (Oliveri, Calì and Catalano 2002) the authors studied the dynamics of the motorcycle, using multibody modelling in which the rider was considered as an immobile equivalent mass.

To calculate the exact value of the polar moments of inertia of the wheels complete with accessories, reported with the values of mass supplied by the manufacturer in Table 1, it was necessary to construct MCAD models.

Table 1: Inertial characteristics of wheels

Front wheel		***Rear wheel***	
Mass	12 kg	**Mass**	14 kg
Moment of inertia	0,5 kg m^2	**Moment of inertia**	0,7 kg m^2

The wheels were modelled with particular accuracy due to the fact that they have a predominant influence on the dynamics of the motorcycle. In fact, their moment of inertia effects both acceleration and braking, the gyroscopic effect determines the stability of the motorcycle and, as suspended masses, they contribute to the correct function of the suspension units.

The present study investigates the effect produced by the movement of the rider's body on the achievable performance of the motorcycle. With a view to developing an actively controlled motorcycle model, an anthropomorphic schematic of a mobile rider was introduced in order to analyse its interactions with the motorcycle and its influence on the overall dynamics when travelling along a rectilinear path and moving through a curve.

2. MODELLING OF RIDER-SYSTEM

NURBS Rhinoceros modelling software was used to construct a parametric model of the human body, which could then describe any subject by inserting his anthropometric measurements in an apposite form.

The model of the rider meets the following requirements:

1. opposes dynamic forces when the motorcycle performs particular manoeuvres;
2. moves in agreement with reality;
3. minimises the calculation time due to the increased number of degrees of freedom.

The model of the rider consists of 15 parts plus eight parts of negligible mass which were inserted in order to be able to correctly position and centre the constraints to which they refer.

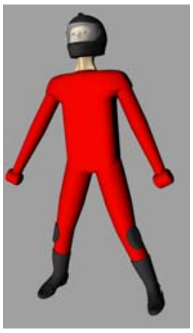


Figure 1: Model of rider constructed using Rhinoceros

Overall, the rider has 24 d.o.f., of which five are controlled in the more demanding manoeuvre (moving through a curve), while the remainder are activated indirectly. The choice of the types of constraints at the joints was principally based on the study of the kinematics activated by the rider during the execution of the manoeuvres.

The type of constraint between the joints making up the model are as follows (table 2):

Table 2: Kinematic constraints between the parts

PART	*NECK*	*SHOULDER*	*ABDOMEN*	*ELBOW*	*HAND*
HEAD	F		--	--	--
CHEST	F	S	R	--	--
ARM	--	F	--	F	--
FOREARM	--	--	--	R	R
PELVIS	--	--	R	--	--
THIGH	--	--	--	--	--
SHIN + FOOT	--	--	--	--	--
HANDLEBAR	--	--	--	--	R

PART	*BUTTOCK*	*KNEE*	*PEDAL*	*SADDLE*
HEAD	--	--	--	--
CHEST	--	--	--	--
ARM	--	--	--	--
FOREARM	--	--	--	--
PELVIS	F	--	--	P
THIGH	S	F	--	--
SHIN + FOOT	--	R	S	--
HANDLEBAR	--	--	--	--

Where: F= Fixed joint, R= Revolute joint, P= Planar joint S=Spherical joint

The postural angles were chosen after having measured, on the real motorcycle, the relative distances between the saddle, the handlebar and the footrest, and their respective spatial orientations.

Naturally, these "cyclistic" dimensions do not identify a unique possible posture. Figure 2 shows three views of the characteristic angles considered for the arrangement of the rider's limbs and body.

This represents the default configuration used as a starting point for considering the movement of different body parts during the simulations. The orientation of rider constraints with respect to the ground and the type are shown in the table 3, where φ_x φ_y φ_z are the angles measured in relation to the ground.

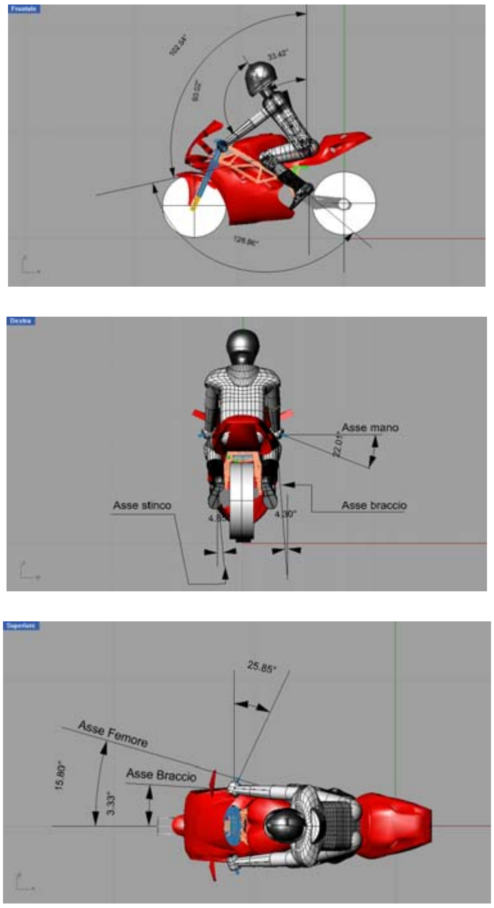


Figure 2: Arrangement of the axes of rider's body parts

Table 3: Orientation of rider constraints with respect to the ground

CONSTRAINT	TYPE	$\varphi_X,\varphi_Y,\varphi_Z$
HEAD-NECK	FIXED	
NECK-CHEST	SPHERICAL	5,-40,-4
SHOULDER-CHEST (SX)	SPHERICAL	0,0,0
SHOULDER-CHEST (DX)	SPHERICAL	0,0,0
ARM-FOREARM (SX)	REVOLUTE	-80,3,-43
ARM-FOREARM (DX)	REVOLUTE	80,3,43
CHEST-ABDOMEN	SPHERICAL	0,-35,6
ABDOMEN-PELVIS	REVOLUTE	0
PELVIS-THIGH (SX)	SPHERICAL	0,0,0
PELVIS-THIGH (DX)	SPHERICAL	0,0,0,
THIGH-SHIN (SX)	REVOLUTE	-70,-20,174
THIGH-SHIN (DX)	REVOLUTE	-70,20,-174

Exploiting the laws of the anthropomorphic model Hybrid III, the dynamic characteristics of the joint can be modified through a single parameter. This allowed the model to be managed more easily and, above all in the first phase, made it possible to understand which joints play an important role in the problem.

The constraints between the modelled rider and motorcycle are:

- the grip of the hands on the handlebar (bushing)
- the feet placed on the footrests (spherical joint)
- the contact between the chest and the saddle-fuel tank unit (contact force)

- the contact between the inside of the thigh and the sides of the fuel tank (contact force).

2.1. Lower limbs

Particular attention was paid to the lower limbs, given that these constitute a fundamental element for the lateral dynamics of the rider. It was observed that the axis of the pelvis and the line linking the knee joints tend to remain parallel during manoeuvres. The legs of the anthropomorphic model were, therefore, considered as a kinematic motion which can be used to guide the transverse movement of the body.

For this reason, the lower limbs were schematised as a deformable quadrilateral in the space constituted by 5 rigid bodies and 4 revolute-joints (Fig. 3):

- 2 parts represent the foot-calf (blue);
- 1 part represents the pelvis (yellow);
- 2 parts represent the thighs (red).

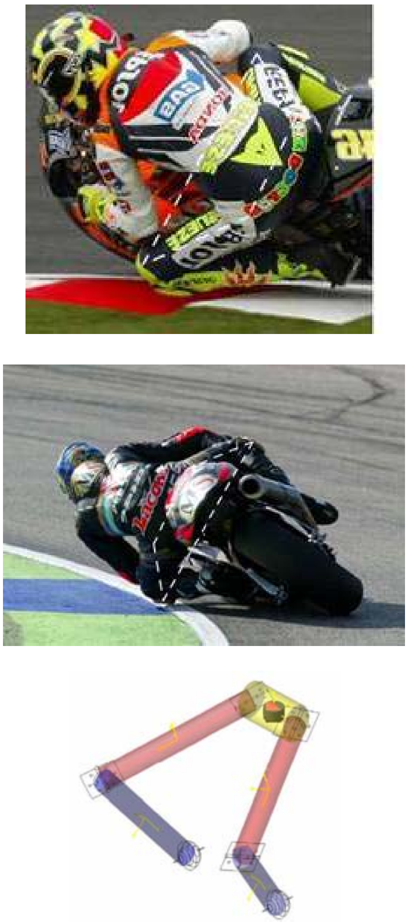


Figure 3: Schematisation of lower limbs

The analyses conducted on the motorcycle using ADAMS/View software showed that the lateral displacement of the rider can be reproduced effectively using this kinematic motion.

With the adoption of these constraints, it was possible to define an efficient schematic with 21 degrees of freedom.

2.2. Forearm-hand

When moving, the rider acts on the handlebar/steering column to correct the trajectory and maintain the desired angle of roll, and his hand and wrist are, therefore, particularly mobile. His wrist must, in fact, second his movements both when he has to “lean into the bend” on entering a curve and when his barycentre returns within the motorcycle’s plane of symmetry on exiting. To this end, two were used, positioned at the hand’s centre of mass. The first revolute joint, connecting the hand to the handlebar and the constraint along the axis coincident with this, allows the rotation of the forearm at the wrist due to the rider's pitching movements when he passes from a prone to supine position and *vice versa*. The second revolute joint, whose axis is orthogonal to the first, allows rotation between the wrist and forearm when the rider's body moves laterally.

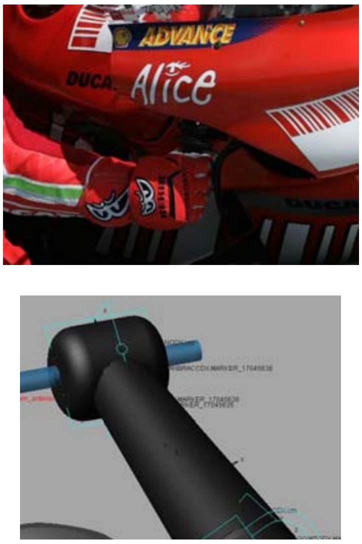


Figure 4: Forearm-hand constraint and hand-handlebar contact

2.3. Forearm-arm

For this constraint it was considered appropriate to use a revolute joint. In reality, the arrangement of the revolute joint axis is also influenced by the rotation of the wrist along the axis of the forearm. However, for simplicity, this aspect was ignored here since it is not relevant to the control of the motorcycle. The hinge axis was fixed with an inclination of 48° with respect to the handlebar axis.

2.4. Mass of individual parts

The inertia characteristics of the single parts making up the rider's body were deduced from the WORLD SID (World Side Impact Dummy) software program, which refers to the ISO/TC22/SC12/WG5 standard describing the anthropometric characteristics of an average-sized automobile driver on the basis of the data provided by AMVO (Anthropometric Specification For Mid-Sized Male Dummy). Table 4 shows the anthropometric characteristics of the various body parts and the position of their centres of mass, calculated with respect to the ground. Table 5 gives the principal moments of inertia of these body parts.

Table 4: Mass characteristics of the rider's body parts and their position with respect to the ground

PART	MASS [Kg]	DENSITY [kg/m^3]	VOLUME [m^3]	CM (x,y,z) [m]
HEAD + HELMET	5.639	1215.1581	0.004641	-0.772, 0.0007, 1.534
NECK	0.965	1347.0922	0.000716	-0.729, -0.0007, 1.399
CHEST	23.763	871.7488	0.027259	-0.606, -0.0007, 1.228
ABDOMEN	2.365	230.8081	0.010247	-0.449, -0.0008, 0.997
PELVIS	11.4	1320.8448	0.008631	-0.357, -0.0003, 0.865
SHOULDER	0.001	1.2434	0.000804	-0.641, 0.222, 1.272
ARM	1.769	757.3291	0.002336	-0.754, -0.228, 1.183
FOREARM	1.769	757.3291	0.002336	-0.754, 0.228, 1.183
ELBOW	0.001	1.8290	0.000547	-0.984, 0.241, 1.005
KNEE	0.001	0.8834	0.001132	-0.743, -0.218, 0.770
HAND	0.487	825.8475	0.000590	-1.153, -0.252, 0.876
THIGH	8.614	883.7219	0.009747	-0.514, -0.152, 0.821
BUTTOCK	0.001	0.3257	0.003070	-0.514, 0.152, 0.821
SHIN + BOOT + FOOT	5.57	807.2962	0.006900	-0.545, -0.203, 0.586
Total weight and global barycentre coordinates	80.558			-0.57,-2.68E-004,1

Table 5: Inertia characteristics of the body parts and global inertia of the rider with respect to his centre of mass

PART	Ixx, Iyy, Izz [kg m^2]
HEAD + HELMET	-0.772, 0.0007, 1.534
NECK	-0.729, -0.0007, 1.399
CHEST	-0.606, -0.0007, 1.228
ABDOMEN	-0.449, -0.0008, 0.997
PELVIS	-0.357, -0.0003, 0.865
SHOULDER	-0.641, 0.222, 1.272
ARM	-0.754, -0.228, 1.183
FOREARM	-0.754, 0.228, 1.183
ELBOW	----
KNEE	----
HAND	5.935E-004, 5.935E-004, 4.672E-004
THIGH	0.146, 0.416, 0.031
BUTTOCK	----
SHIN + BOOT + FOOT	0.1501, 0.1423, 0.01647
Overall inertia with respect to the CM	8.108, 8.735, 4.264

3. THE MOTORCYCLE CONTROL SYSTEM

The control of the motorcycle was effected adopting a system consisting of three modules: one proportional, one derivative and one a function of the second derivative, regulating the turning couple applied to the steering column and the torque applied to the rear wheel. In order to guarantee the correct kinematic behaviour of the motorcycle, the control system must be able to perform four fundamental functions:

- Impose the angle of roll as a function of the curvature of the trajectory and the velocity of the motorcycle
- Stabilise the falling motion of the motorcycle (CAPSIZE)
- Correct the trajectory
- Correct errors in the velocity

Figure 5 shows a schematic diagram of the input and output parameters of the motorcycle control system.

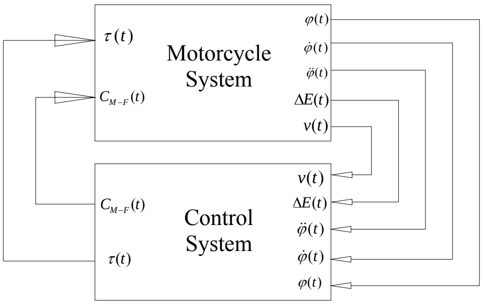


Figure 5: Block diagram of the control system

3.1. Control applied to the steering column

The equation regulating the directional behaviour of the system is:

$$\tau(t)=K_1(\varphi(t)-\varphi_{ideal}(t))+K_2(\dot{\varphi}(t)+K_3\ddot{\varphi}(t)+K_4\Delta E(t) \qquad (1)$$

where τ(t) is a couple applied to the steering column, φ(t) is a value of the motorcycle's roll angle at instant t, $\varphi_{ideal}(t)$ is the desired value at the same instant, expressed by:

$$\varphi_{ideale}=arctg\frac{R_g\cdot\Omega^2}{g}+arcsen\frac{t\cdot sen(arctg\frac{R_g\cdot\Omega^2}{g})}{h-t} \qquad (2)$$

and ΔE(t) is the distance from the reference trajectory. Expression (1) is that of a PD type controller with the addition of the term relative to the acceleration of roll. The function of the first and second derivatives of the roll angle is to allow the introduction of the concept of predicting the roll motion: i.e. it is supposed that the rider is able to perceive the magnitude of the variations in the roll angle and, therefore, the values of $\dot{\varphi}(t)$ and $\ddot{\varphi}(t)$. With this system, the controller applies a steering couple until the roll angle is equal to that desired, thereby zeroing the second element of the equation. Once the roll motion is stabilised, the couple at the steering column is still not zero due to the term relative to the deviation from the reference trajectory ΔE(t) and the controller therefore operates to compensate for this error by continuing to act on the steering column. It should also be noted that the difference between φ(t) and φ_{ideal}(t) initially assumes a negative value and this allows the simulation of the real behaviour. During the phase of entering a curve, in fact, every rider, whether consciously or not, tends to steer from the opposite side of the curve in order to generate a centrifugal force which provokes a "tipping over" towards the inside of the curve.

The variables determining the couple on the steering column are shown in Figure 6.

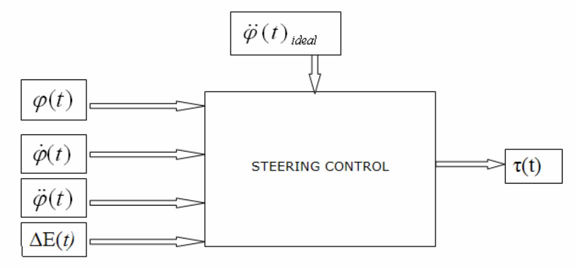


Figure 6: Schematic diagram of steering control

In the model developed in ADAMS/View, the couple at the steering column is introduced using an "SFORCE" (appendix A).

It can be seen that the controller for the calculation of the steering couple does not refer to the roll angle of the motorcycle, but to that of the global motorcycle-rider system. In this way is possible to take into consideration displacements in the rider's barycentre and to observe the effects of these on the dynamic parameters. The IF function multiplying the variables of the system allows the controller to intervene on the steering column only when the velocity of the motorcycle has reached a value which ensures its stabilisation (CAPSIZE) (Cossalter 1997).

In the equation (1) the four coefficients K_1, K_2, K_3 and K_4 together determine the trajectory prediction times and the "driving style" of the virtual rider. In particular, varying the coefficients one by one, produces the the following qualitative effects:

***Increasing K_1*:** the system reacts more rapidly to perturbations in its state of equilibrium, advantageous in the case, for example, of a lateral disturbance caused by a gust of wind. There is, however, an increase in the oscillation of the roll motion around the equilibrium position which is disadvantageous when assuming the roll angle required for a given curve. In this case, too great an increase in K_1 results in exceeding the desired roll angle φ and this can lead to instability if the curve is particularly demanding.

***Increasing K_2*:** there is greater damping of the roll motion oscillation around the equilibrium configuration, making the system more stable. Also here an excessive increase must be avoided or the system will again tend towards instability.

***Increasing K_3*:** the system reacts more quickly to variations in the curvature of the trajectory, attaining the required roll angle more rapidly. The frequency of the roll motion oscillation increases as does its amplitude around the equilibrium position. Further increase in K_3 produces a second, higher frequency of the roll motion oscillation, making the system instable.

***Increasing K_4*:** the system is more sensitive to corrections to errors of position and trajectory.

4.2 Control of velocity

In order to control the velocity of the motorcycle, torque was applied to the rear wheel. At running speed, this torque is described by the function:

$$C(t) = C_0 + K_5(V_{motorcycle}(t) - V_{desired}(t)) \qquad (3)$$

The term C_0 has the function of accelerating the motorcycle, thereby regulating the transient. The proportional module manages the transient and, therefore, regulates the velocity of the motorcycle.

4. LOCALISATION OF THE MODEL'S BARYCENTRES AND IMPLEMENTATION OF THE ALGORITHM CONTROLLING THEIR DISPLACEMENT

As noted above, the rider's movements influence the dynamic parameters of the motorcycle during the execution of some manoeuvres. For this reason, it is important to localise the position of the centres of mass of the motorcycle, of the rider and of the overall system, and to monitor their displacements.

Calculating the global barycentre of the motorcycle-rider system, using the classic equation of Rational Mechanics (Oliveri 1993), six general constraints (GCON), one for each direction of the centre of mass, were introduced into the ADAMS model:

Fictitious part, roll angle allowing the calculation of the roll angle referring to the position of the global barycentre through the expression:

$$\varphi = a\tan\frac{S_y}{S_z} \qquad (4)$$

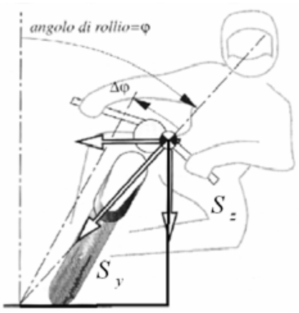


Figure 7: Calculation of overall roll angle

Fictitious part, pitch angle allowing the calculation of the pitch angle.

Sensor part, trajectory has the task of moving along the trajectory and continuously calculating the curvature (5) and, therefore, the ideal roll angle on the basis of equation (2). It also allows the calculation of the motorcycle's distance from the required trajectory.

$$C(t) = \left| \frac{a_n(t)}{V_x^2(t)} \right| \qquad (5)$$

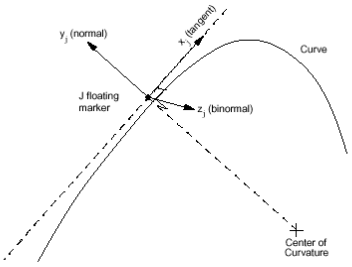


Figure 8: Alignment of marker tangent on curve

The syntax of the CURVATURE function is defined in ADAMS/View by expression in appendix B.

Analogously, two further fictitious parts were used in order to evaluate **roll velocity** and to apply **aerodynamic forces.**

A sixth power polynomial was used to describe the trajectory, providing the best compromise between oscillations of the curve and approximation of the points of control. This approach allowed an active control of the trajectory. The trajectory sensor reads the curve with a certain anticipation, chosen in function of the velocity, and the system (virtual rider) can choose the control parameters, operating on both motorcycle and body movement to obtain the optimal trajectory.

5. ANALYSIS OF RESULTS

The results of the simulations performed were analysed in order to determine the way in which the movements of the rider (Fig. 9) effect the dynamic parameters of the motorcycle.

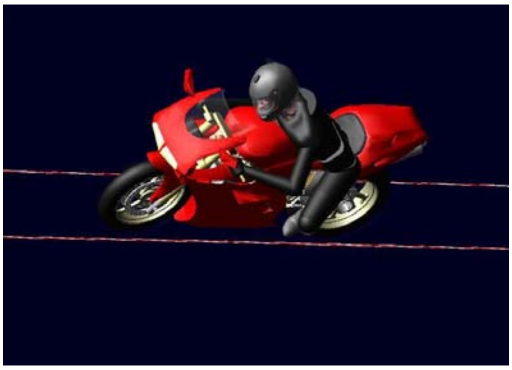


Figure 9: Model of motorcycle with rider moving through a curve

5.1. Motion through a curve

Over a simple track consisting of rectilinear entry path – constant radius curve – rectilinear exit path at constant velocity, it is possible to identify 5 phases: a condition of uniform rectilinear motion; the transient of entering the curve; a stationary condition reached within the curve; the transient of existing the curve; a final condition of uniform rectilinear motion.

Of these, the phases important for the lateral dynamics are those of the stationary state in the curve and the two transient states occurring between rectilinear motion and passing through the curve.

The characteristic phenomenon of the latter phase greatly depend on the driving style of the rider, i.e. on the characteristics of the control system. For this reason, three different rider positions are analysed below: fixed, displaced towards the inside of the curve and displaced towards the outside of the curve.

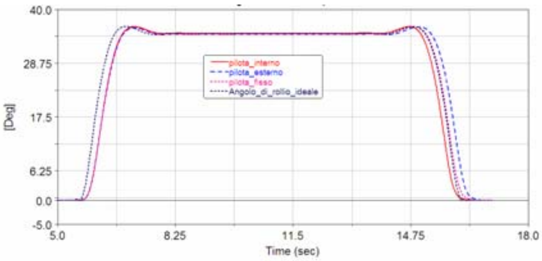


Figure 10: Roll angle relative to the global barycentre

The trend of the roll angle relative to the global barycentre makes it possible to assess the inclination of the motorcycle-rider system in the various sections of the track. In all three cases, this inclination coincides with the reference roll angle, except for a small error which becomes larger in the transients of entering and exiting the curve. No significant variations were observed during the phase of entering the curve, because of the rider anticipates this phase with his movements and the control system therefore has the time to compensate for the error. On exiting the curve, instead, the righting of the motorcycle is anticipated or delayed when the rider is, respectively, displaced towards the inside or outside of the curve.

The movement of the rider towards the plane of symmetry, in fact, favours the righting of the motorcycle in the first case and opposes it in the second.

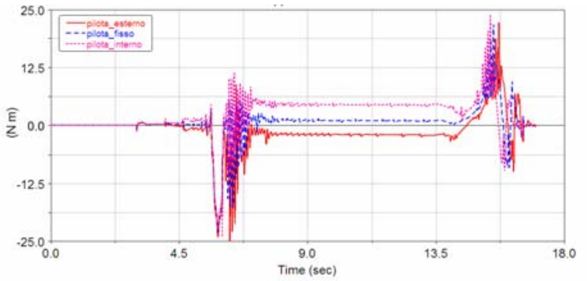


Figure 11: Torque applied to steering column

The turning couple at the steering column shows the continuous corrections made by the rider during the phases of entering and exiting the curve. It is clear that, to maintain the roll angle while passing through the curve, the rider exerts a greater action on the steering column when he moves towards the inside of the curve due to an increase in the moment of roll generated by the weight force. This requires him to act more strongly against "tipping over" by steering harder towards the inside of the curve.

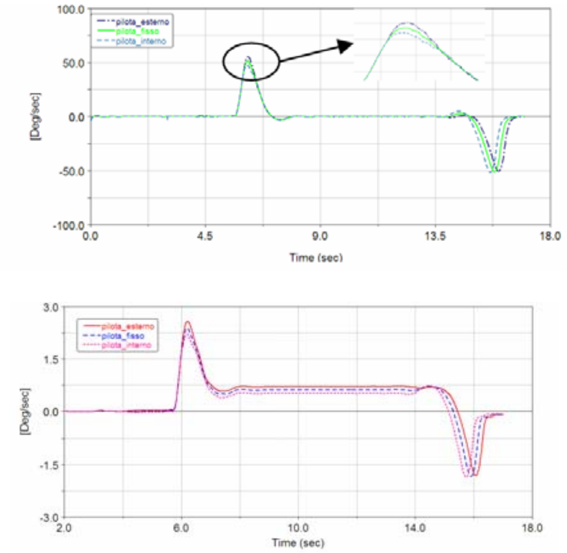


Figure 12: Velocity of roll and pitch

The trend of the motorcycle's roll velocity shows the presence of two peaks at entering and exiting the curve due to the greater action applied to the steering column during these phases. When running at speed, the motorcycle is stabilised and the roll velocity is zero. The offset of the peaks in the phase of entering the curve relative to the three cases should also be noted. This shows how, given parity of the time required to reach the reference roll angle, the movement of the rider towards the inside of the curve necessitates a slower roll velocity. Also the pitch velocity is reduced when the rider moves towards the inside of the curve in both the phase of entering the curve and to that of passing through it. However, since this is not a controlled parameter, it is possible to imagine that the system reacts somewhat slowly to pitching, probably due to the lesser inclination of the motorcycle and two lower value assumed by the lateral forces of adherence which, together with the centrifugal forces acting on the barycentre, are responsible for the moment of pitch.

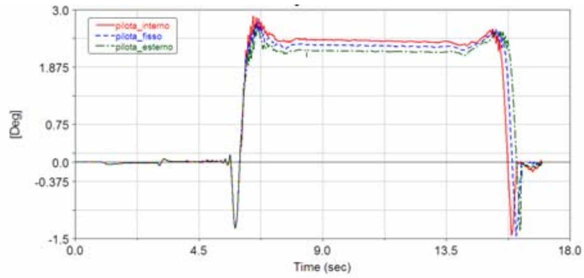


Figure 13: Angle of steering column

The trend of the steering angle allowed an assessment of the low values of the forces acting on the steering column. This made it possible to ignore the contribution due to the gyroscopic moments of the wheels with increasing roll angle. In the phase of entering the curve, there is a peak with a negative sign, representing the initial phase of "countersteering" which permits the insertion into the curve.

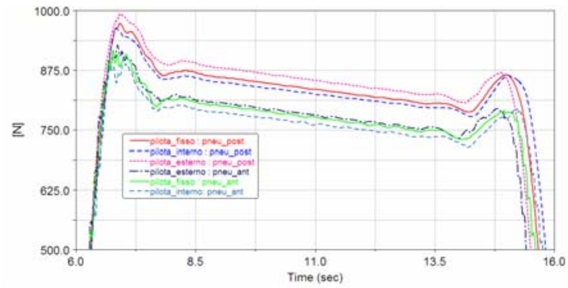


Figure 14: Comparison between the lateral forces acting on the tyres

Figure 14 shows a comparison between the three lateral reactive forces acting on the tyres. It can be seen that the rear wheel exerts greater adherence. Further, in the case of the rider displaced towards the inside of the curve, the lateral force is reduced. This allows the rider to delay the phenomena of slipping associated with the loss of adherence at the tyre-road surface which, given parity of trajectory curvature, will occur at higher velocities. The downward trend of the curve is due to the motorcycle deviating slightly from the intended trajectory.

Thus, the lateral displacement of the anthropomorphic model made it possible to take greater advantage of the characteristics of the tyres which was then translated into an increase in the maximum velocity possible along the track.

6. CONCLUSIONS

Based on co-simulation, the methodology used allows the parametric multibody modelling of a motorcycle with an anthropomorphic model of the rider. With this model, it is possible to evaluate the connection between the movement of the rider's body and that of the overall barycentre of the motorcycle. The approach to the problem of motorcycle dynamics described here is substantially innovative, in that it was complemented by and integrated with a biomechanical component. Numerous configurations were examined to determine the arrangement of the rider/motorcycle constraints and, above all, for the articulated model of the rider, arriving at the definition of an effective scheme with 21 degrees of freedom.

The simulations performed demonstrated the plausibility of the anthropomorphic model in terms of kinematic behaviour, interaction with the motorcycle, and biomechanics given that, even under the most demanding conditions, the physiological limits of joint mobility were always respected.

From the point of the achievable performance of the motorcycle, introducing the anthropomorphic model of the rider resulted in evident improvements, in particular when travelling through a curve.

The rider model developed here is simple and easily managed and can, therefore, be considered a valid starting point for a future implementation of dedicated control systems.

The simulations revealed two degrees of freedom which, due to the kinematic characteristics of the model developed here, are particularly important: rotation in the sagittal plane of the elbow and the knee.

An active type of modelling would allow a more effective compensation for effects of the inertia forces during the phases of acceleration and braking. This would make it possible to define a more realistic distribution on the various constraints of the forces exchanged between the rider and the motorcycle.

ACKNOWLEDGMENTS

The authors wish to thank Dott. Ing. Simone Di Piazza and Dott. Ing. Marco Paradisi of Ducati Motor Holding S.p.A. for the data regarding the motorcycle and the rider and Dott. Ing. Daniele Catelani of MSC/software for his collaboration.

APPENDIX

A)

```
"IF(TIME3:0,0,1)*((850*(VARVAL(.model_deformabi
le_.ANGOLO_DI_ROLLIO_COMPLESSIVO)-VAR
VAL-(.model_deformabile.Phi_id_tot)))+VARVAL
(.model_deformabile.VEL_ANGOLO_ROLLIO_COM
PLESSIVO)+VARVAL(.model_deformabile_.ACC_A
NGOLO_DI_ROLLIO_COMPLESSIVO))+VARVAL(
.model_deformabile.ERRORE_DI_POSIZIONE))".
```

B)

```
ABS((ACCY(.MARKER_MOBILE_SU_TRAIETTOR
IA,MARKER_MOBILE_SU_TRAIETTORIA)+0.0000
001)/(VX(.MARKER_MOBILE_SU_TRAIETTORIA,.
mar_rif,.MARKER_MOBILE_SU_TRAIETTORIA)**
2+0.0000001)).
```

REFERENCES

Bradley, J., 1996. *The racing motorcycle.* Broadland Leisure Publications, ISBN 0 9512929 2 7.

Cossalter V., 1997. Cinematica e dinamica della motocicletta. Padova: Edizioni Progetto.

Funke, J., Breuer, B., Landau, K., 2000. Determination of connective Forces between Rider and Motorcycle - an Approach to Men-Machine-System Investigation. *Proceedings of the 2000 International Motorcycle Safety Conference,* Munich, Germany, September 2000.

Imaizumi, H., Fujioka, T., Omae, M., 1996. Rider model by use of multibody dynamics analysis. *Japanese Society of Automotive Engineers*, Vol. 17, No.1, 75–77.

Yokomori, M., Higuchi, K., Ooya, T., 1992. Rider's operation of a motorcycle running straight at low speed. *JSME International Journal*, Vol.35 No. 4 553-559.

Oliveri, E., 1993. *Lezioni di meccanica razionale.* Catania Italy CULC.

Oliveri S.M., Calì M. Catalano L., 2002. Dynamics of Motorcycle using flexible elements. Proceedings of the International Design Conference, pp. 1227-1236, May 14 - 17, 2002, Dubrovnik.

Pacejka, H.B., 2002. *Tyre and Vehicle Dynamics*. Butterworth-Heinemann, ISBN 0 7506.

Rice R.S., 1978. Rider Skill Influences on Motorcycle Maneuvering. *SAE* N° 780312 Warrandle PA.

Wu, J.C., Liu T. S., 1995. Fuzzy control of rider-motorcycle system using genetic algorithm and auto-tuning. *Mechatronics*, Vol. 5, No. 4, 441-455.

AUTHORS BIOGRAPHY

Michele Calì was born in Brindisi, on 19 Aug 1969. In 1974 he went to Catania were he was educated, and in 1996 he was graduated in Mechanical Engineering. In 2001 he achieved the PhD of Mechanic of Structures in Mechanical Engineering Department of the Catania University. Actually he is in Postdoc position and worked, as researcher, with Professor Massimo Oliveri, at the Design Technical researched group. He works in some projects with Ferrari S.p.A., Ducati Motorcycles Holding S.p.A., CIRA S.p.A. (Italian Aerospace Researched Centre), Centro Ricerche FIAT (CRF), Piaggio V.E. S.p.A. and Erg petrol S.p.A. He works in research on dynamic simulation and deformation studies of engine, timing system and mechanical system in generally. He takes part in European research projects, particularly in the research on engine 3D modelling. His 12-years research on mechanical system and design has documented by numerous papers, lectures and articles.

Massimo Oliveri, professor of Design and Method of the Industrial Engineering in Engineering Faculty of Catania. Professor of Computer Assisted Drawing to the same University in the degree in Mechanical Engineering.

He has developed, in the course of his academic career, scientific activities in different sectors of the Machine Design that get from strain and stress analysis, to CAD in mechanical design, to fatigue in welds, to study of complex kinematics using multi-body software. He has manage and cooperate to the research activity with some Italian University, with C.R.F. in Orbassano and ELASIS in Pomigliano d'Arco and with some Italian industries (Ferrari S.p.A., Ducati Motors Holding S.p.A., Fiat Auto and Piaggio V.E. S.p.A.)

Gaetano Sequenzia was born in Catania (Italy) on 19 January 1980, graduated at the University of Catania in 24 July 2004 in Mechanical engineering with bachelor ballot 110/110 cum laude.

Actually Philosophy Doctor (Mathematic engineering) at University of Catania. Recent technical interests regard dynamic simulations of mechanical components, as crankshaft cranktrain, valvetrain and chains.

Francesco Trovato was born in Catania (Italy) on 5 February 1981, graduated at the University of Catania in 24 July 2007 in Mechanical engineering. Actually he work at Moto Guzzi S.p.A.

MODELLING, SIMULATION AND ERGONOMIC STANDARDS AS SUPPORT TOOLS FOR A WORKSTATION DESIGN IN MANUFACTURING SYSTEM

Antonio Cimino[(a)], Giovanni Mirabelli[(b)]

[(a) (b)] Modeling & Simulation Center - Laboratory of Enterprise Solutions (MSC-LES)
Mechanical Department, University of Calabria, 87036 Rende (CS), Italy

[(a)] acimino@unical.it, [(b)]g.mirabelli@unical.it

ABSTRACT

It is the intent of the work to achieve the ergonomic effective design for a workstation belonging to a manufacturing system operating in the field of mechanical parts production. To this end, the NIOSH 81, NIOSH 91, Burandt Schultetus and OWAS analysis are accomplished for developing an improved workstation configuration in terms of interaction between humans and their working environment. Moreover the authors adopt an advanced approach based on Modeling & Simulation (M&S) in order to implement a three dimensional environment for recreating with satisfactory accuracy the real workstation and for detecting ergonomic problems that otherwise would be difficult to detect. In conclusion a final workstation configuration is proposed and significant ergonomics improvements are achieved.

Keywords: workstation effective design, ergonomic standards, modeling, simulation.

1. INTRODUCTION

An overview of the state of the art, starting from the second half of the 1990s, reveals that manufacturing systems continuously provide challenging problems for researchers and scientists working in the applied ergonomics field. A number of different research works and scientific approaches have been proposed, trying to achieve an ergonomic effective design of the workstations belonging to a manufacturing system.

In the late ’90, the ergonomic effective design was mostly supported by videotaping systems used for data collection, i.e. videotape of the work method being used in the workplace (Das and Sengupta 1996; Scott and Lambe 1996; Engstrom and Medbo 1997; Vedder and Hellweg 1998; Kadefors and Forsman 2000; Neumann et al. 2001).

In addition, several authors propose an approach based on a single ergonomic performance measure (i.e. lift index, energy expenditure measure, work postures etc.) due to the application of a specific ergonomic standard such as the Ovako Working Posture analysis System (OWAS), the Burandt-Schultetus analysis, the NIOSH 81 and NIOSH 91 equations (NIOSH stand for National Institute for Occupational Safety and Health), the RULA method (RULA stands for Rapid Upper Limb Assessment), the Garg analysis. Further information about the cited ergonomic standards can be found in the Niosh Technical Report 81-122 (1981), the Scientific Support Documentation for the Revised 1991 NIOSH Lifting Equation (1991), Waters et al. (1994), Kharu et al. (1981), Schultetus (1980) and Garg (1976), McAtamney and Corlett (1993). Examples of research works based on a single ergonomic performance measure are reported in Carrasco et al. (1995), Grant et al. (1997) Van Wendel de Joode et al. (1997), Mital and Ramakrishnan (1999), Temple and Adams (2000), Lin and Chan (2007), Waters et al. (2007).

In order to achieve relevant ergonomic improvements some authors propose an effective ergonomic design based on the integration of different ergonomic standards. Wright and Haslam (1999) investigate manual handling risks within a soft drinks distribution centre using the OWAS postural analysis and the NIOSH equations. Russell et al. (2007) compare the results of different ergonomic standards for evaluating ergonomic risks in lifting operations. Jones et al. (2005) present an examination of three common pub occupations (bartending, waitressing and cooking by means of RULA method and NIOSH Lifting Equation.

Another critical issue, other than the choice of the ergonomic performance measure (i.e. single or multiple) or the methods used for data collection (i.e. video type system), is whether the ergonomic standards have to be applied directly in the real system. Usually the application of the ergonomic standards in the real system requires huge amount of money and time for testing all the workstations configurations, work assignments, work methods, etc. Consequently, researchers and practitioners very often use Modelling & Simulation (M&S) as decision support tool for choosing correctly, for understanding why, for diagnose problems and explore possibilities (Banks, 1998).

Over the years the M&S approach has became more and more appealing thanks to the numerous advantages such as the possibility to study ergonomic issues at the earliest stages of design in order to avoid potential future ergonomic redesign in the real-world system.

Feyen et al. (2000) propose a PC-based software program (based on the integration of a Three-Dimensional Static Strength Prediction Program, 3DSSPP, for biomechanical analysis with a widely used computer-aided design software package, AutoCAD). As consequence, the authors are able to study ergonomic issues during the design phase taking into consideration different design alternatives.

Chang and Wang (2007) propose a method for conducting workplace ergonomic evaluations and re-design in a digital environment with the aim of preventing work-related musculoskeletal disorders during assembly tasks in the automotive sector.

Longo et al. (2006) use M&S in combination with ergonomic standards and work measurement for the effective design of an assembly line still not in existence. The authors propose a multi-measures approach with the aim of obtaining a different work assignment to each workstation, better line-balancing and better ergonomic solutions.

Santo et al. (2007) propose an ergonomic study on working positions in a manufacturing company (by using the simulation software eM-Workplace) and providing, as result, remarkable ergonomic improvements.

The main goal of the paper is to achieve the ergonomic effective design of a real workstation belonging to a manufacturing system operating in the field of mechanical parts production. To this end, several ergonomic standards are accomplished for developing an improved workstation configuration in terms of interaction between humans and their working environment. Moreover the authors adopt an advanced approach based on Modelling & Simulation (M&S) in order to implement a three dimensional environment for recreating with satisfactory accuracy the real workstation and for detecting ergonomic problems that otherwise would be difficult to detect.

Before getting into details of the study let us give a brief overview of each section of the paper. Section 2 describes the manufacturing process. Section 3 gives specific details on the simulation model implementation and validation. Section 4 presents the ergonomics standards accomplished for the effective ergonomic design. Section 5 presents the simulation results. Section 6 shows the effective ergonomic redesign of the workstation under consideration (final configuration). The last section reports the conclusions that summarize the scientific contribution of the work and the research activities still on going.

2. THE MANUFACTURING PROCESS

The research work focalizes on the ergonomic effective design of a workstation belonging to a manufacturing system specialized in the field of mechanical parts production. The manufacturing plant is located in the South part of Italy and covers a surface of 11000 square meters. It manufactures a mechanical component for agricultural machineries engines on behalf of third parties and consists of three different areas:

- The raw materials storehouse: a basic mechanical component is stored in pallets loaded into pallet racks. The pallets and the mechanical components are moved by means of a system of automated conveyors, automated storage and retrieval machines. The storage area is 11 meters high and covers a surface of 2000 square meters;
- The production area: it consists of three workstations: the flange workstation, the screws workstation and the assembly workstation.

  The flange workstation products the flanges being assembled to the mechanical component stored in the raw materials storehouse. It employs 4 operators and contains 4 motorized ball control lathes characterized by the same productive process. The workstation layout covers a surface of 2500 square meters.

  The Screws workstation manufactures 10 different screws types; in particular the hexagonal screws 4 cm high are used for fixing together the mechanical component and the flange, while the other screws types represent, for the manufacturing system, final products directly sold to the customers. The workstation employs 5 operators and it is made by a broaching machine, a slotting machine, a machine for worm screws, a numerically controlled machine for worm screws with robot and a shaving machine. The workstation layout covers a surface of 2500 square meters.

  The Assembly workstation assembles the mechanical component, the flange and the screws in order to obtain the main final product for the enterprise. The assembly activities are performed during all the day by three experts operators (each operator works eight hours per day). The plant layout covers a surface of 2000 square meters.
- The final products warehouse: the assembled workpiece and the screws manufactured in the production department are stored in pallet loaded in pallet racks. As the Raw materials storehouse, such area is completely automated by means of conveyors and automated storage and retrieval machines. The storage area covers a surface of 2000 square meters and is 12 meters high.

Figure 1 shows the whole plant layout of the manufacturing system.

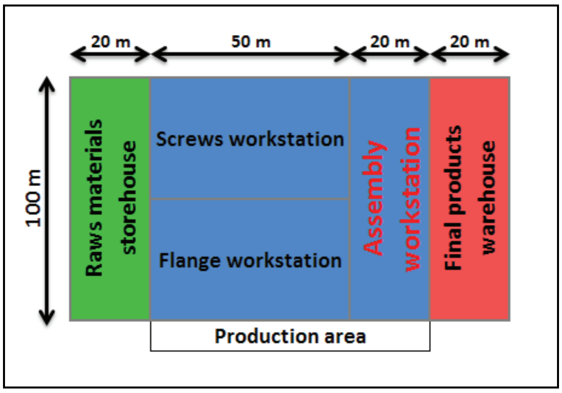


Figure 1: Manufacturing system plant layout

The Assembly workstation represents a bottleneck in terms of ergonomic issues for the company; in fact the operators always complain of strong back pains at the end of the working shift. A preliminary analysis carried out by the company top management shows that the ergonomic problems are related to the weight of the mechanical parts being moved (mechanical component, flange and assembled workpiece), the wrong design of some working equipments and the objects positions into the workstation. In this regards they asked the authors for designing an improved workstation configuration in terms of acceptable ergonomic risk levels.

Let us describe the operations sequence performed in the Assembly workstation: (1) the operator picks up manually the mechanical component placed on a pallet coming from the Raw materials storehouse and puts it on a first work bench; (2) the operator takes a hand-drill placed on the work bench and uses it for drilling the mechanical components five times; (3) the operator puts away the hand-drill, takes the mechanical component and places it on a second work bench 180 cm far away; (4) the operator takes a flange placed on a pallet coming from the Flange workstation and puts it on the second workbench for performing the assembly operations; (5) the operator puts the flange on the mechanical component and makes them adhering by means of a hammer; (6) the operator fixes together the mechanical component and the flange by using three hexagonal screws located in the screws bin coming from the Screws workstation; (7) the operator gets the assembled workpiece and puts it on a pallet placed 670 cm far away; (8) the operator activates a mechanical arm that gets the assembled workpiece and puts it on a conveyor directed to the Final product warehouse.

3. THE SIMULATION MODEL

In this research work the authors present an approach based on the integration of M&S tools and several ergonomics standards (NIOSH81, NIOSH91, Burandt Schultetus, OWAS) for improving the Assembly workstation configuration in terms of interaction between humans and their working environment.

The first step was the development of a simulation model capable of recreating the Assembly workstation with satisfactory accuracy. Before getting into the details of the simulation model development, let us summarize the three main phases characterizing it: collecting data concerning the Assembly workstation (data collection phase), reproducing the real system in the virtual environment from both a geometric and work method point of view (simulation modelling phase) and verifying if the simulation model is an accurate representation of the real system (validation phase).

3.1. Data collection phase

The data collection phase is a mandatory step for developing a simulation model capable of recreating the system under consideration with satisfactory accuracy. In this regards the authors spent a three months period at the Assembly workstation for gathering all the information and data they needed.

Information regarding operator characteristics (age, gender, weight and physical conditions), dimensions (length, height and width), weight and position of all the workstation objects as well as information regarding work method were collected.

The operators characteristics were used for selecting human models capable of representing as much as possible the real operators.

The objects dimensions, weight and position were used for recreating the geometric model of the Assembly workstation.

The work method information were used for reproducing correctly in the virtual environment the workstation assembly process.

3.2. Simulation modelling phase

The simulation modeling phase aims at reproducing the Assembly workstation in the virtual environment from both a geometric and work method points of views. This phase consists of two steps:

1. *Plant layout generation* for recreating the three dimensional geometric model of the workstation;
2. *Human model inserting and training* for reproducing in the virtual environment all the operations characterizing the assembly process.

Section 3.2.1 and section 3.2.2 get into the details of the Simulation Modeling phase main steps.

3.2.1. Plant layout generation

The implementation of the geometric models of the Assembly workstation has been made by using the CAD software Pro-E; in this regards, objects dimensions and weight, gathered in the data collection phase, were inserted into the Cad software as input data for designing geometric models with high level of detail. Table 1 reports description, type, dimensions and weight of the objects mainly used during the assembly operations.

Table 1: Data collection for geometric models

Object desrciption	**Object type**	**Dimensions (cm) L x W x H**	**Weight (Kg)**
Mechanical component	Component	34,5 x 24,5 x 14	14
Hand drill	Equipment	38,5 x 27,5 x 7,5	6
Mechanical arm	Equipment	300 x 200 x 100	80
Conveyor	Equipment	600 x 100 x 50	55
Screws bin	Equipment	30 x 20 15	0,3
Hexagonal wrench	Equipment	15 x 5 x 1	0,2
Hammer	Equipment	23 x 6 x 2	0,6
Screw	Component	4 x 0,5 x 0,5	0,05
Flange	Component	29 x 12,5 x 12,5	9

The plant layout generation has been completed by importing the geometric models into the virtual environment provided by the simulation software eMWorkplace. Note that the geometric models have been located in the same positions the real objects take place in the system in order to recreate exactly in the virtual environment the Assembly workstation.

Figure 2 shows a panoramic view of the virtual assembly workstation.

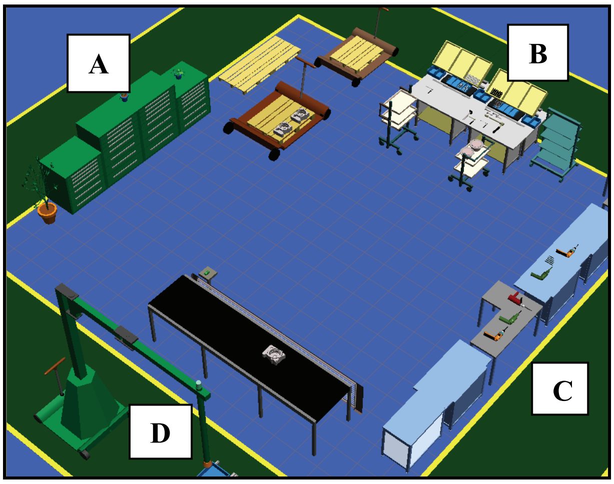


Figure 2: Virtual Assembly workstation

The side "A" consists of some office furniture and the mechanical components placed on the pallets coming from the Raw material warehouse.

The side "B" consists of a workbench, a screws bin, a hexagonal wrench, a hammer and flanges; in this part of the workstation the basic assembly operations are performed.

The side "C" consists of several workbenches and hand drills used for drilling the mechanical component.

Finally the side "D" consists of a mechanical arm, a workbench and a conveyor.

Figure 3 and figure 4 show respectively the main mechanical component and the workbench used for the assembly operations.

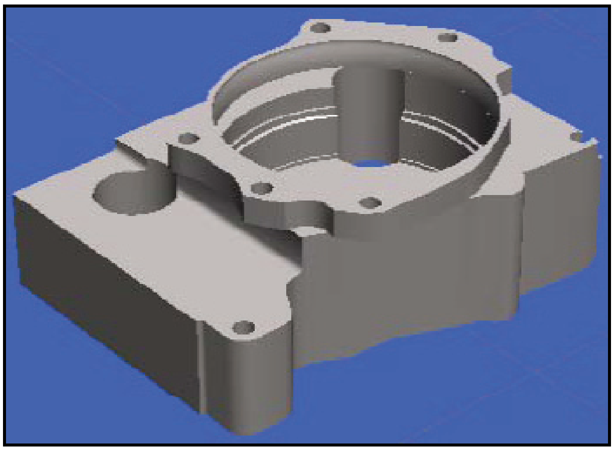


Figure 3: The mechanical component

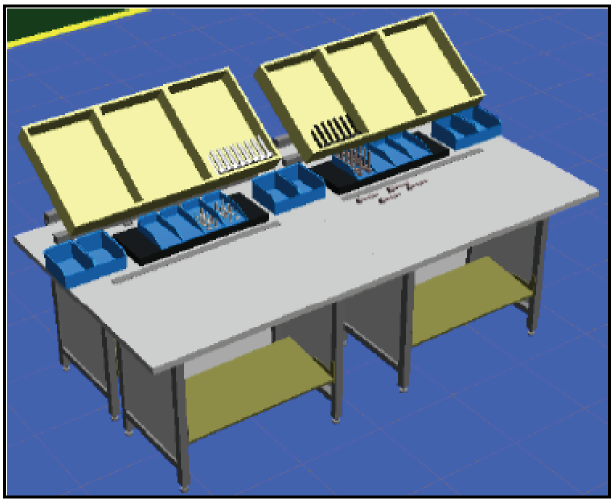


Figure 4: Workbench

3.2.2. Human model insertion and training

After the plant layout generation, humans models insertion and training are needed for reproducing correctly in the virtual environment all the operations performed in the assembly workstation. As reported in the MANUFACTURING PROCESS section, the assembly workstation employs three operators working each one 8 hours per day on different working shift; on this subject in the simulation model development the authors selected from eMWorkplace libraries a human model capable of reproducing as much as possible the real operators characteristics. In fact all the information concerning age, gender, weight and physical conditions of the three operators were averaged out and used as input data for the human model selection.

The table 2 reports the physical characteristics for each operator and their average values as well.

Table 2: Operators physical characteristics

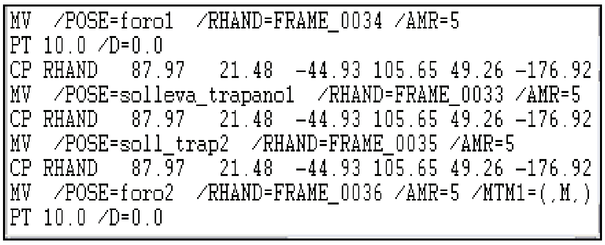
		Operator			
		Op - 1	**Op - 2**	**Op - 3**	**averaged Op**
Physical characteristics	Age	41	44	50	45
	Gender	male	male	male	male
	Height (cm)	175	170	178	174
	Weight (Kg)	76	69	81	75
	Physical Conditions	normal	normal	normal	normal

After the selection of the more suitable human model, it has been inserted in the virtual environment provided by eMWorkplace.

Obviously, at the beginning, the human model is only able to stand in the waiting position; in fact it needs to be trained for reproducing correctly the assembly operations. To this end, eMWorkplace provides the user with a programming language for teaching the basic motions of each operation such as reach, grasp, realize, move, etc.. In this regard the authors have analyzed each operation trying to identify all the basic motions it is made by. Note that such activity has required a huge amount of time due to the high number of basic motions that generally characterizes an operation.

Hundreds of programming code lines have been written for allowing the human model to recreate exactly the real assembly process. Moreover information regarding working postures at the beginning and end of each lifting task, frequency and duration of lifting tasks have been required by eMWorkplace for evaluating the workstation in terms of ergonomic issues. In effect the software allows the user to analyze the simulation model under consideration by means of several ergonomic standards (NIOSH 81, NIOSH 91, Burandt Schultetus, OWAS); on this subject, after the simulation model runs, the user is provided with the ergonomic standards results so that all the workstation ergonomic problems can be identified..

Figure 5 shows several programming code lines written for teaching the human model.



```
MV  /POSE=foro1  /RHAND=FRAME_0034 /AMR=5
PT 10.0 /D=0.0
CP RHAND   87.97   21.48  -44.93 105.65 49.26 -176.92
MV  /POSE=solleva_trapano1  /RHAND=FRAME_0033 /AMR=5
CP RHAND   87.97   21.48  -44.93 105.65 49.26 -176.92
MV  /POSE=soll_trap2  /RHAND=FRAME_0035 /AMR=5
CP RHAND   87.97   21.48  -44.93 105.65 49.26 -176.92
MV  /POSE=foro2  /RHAND=FRAME_0036 /AMR=5 /MTM1=(,M,)
PT 10.0 /D=0.0
```

Figure 5: Some programming code lines

3.3. The validation phase

To increase significantly the probability of success in conducting a simulation study, the validation phase is a mandatory step for determining if the simulation model is an accurate representation of the real system under investigation.

The validation phase has been carried out by means of two different approaches: debugging technique and visualization/animation technique.

The debugging technique is an iterative process whose purpose it is to uncover errors or misconceptions that cause the model's failure and to define and carry out the model changes that correct the errors (Banks,1998). In this regards, during the simulation model development, the authors tried to find the existence of errors (bugs). Given the detected errors, the authors determined the cause of each error and identified the model changes required for their correction. The identified model changes were carried out and further investigations for finding other bugs were accomplished; in effect a change correcting an error may create another one. This iterative process has been continued until no errors were identified in the simulation model.

Visualization/Animation of a simulation model greatly assists the validation phase (Sargent 1996, Bell and O'Keefe 1994). Seeing the animation of the model as it executes and comparing it with the operations of the assembly workstation helped the authors to identify discrepancies between the simulation model and the real system.

Both techniques have been applied with the help of the workstation operators and company production engineers.

The simulation model has been debugged by analyzing all the basic motions of the human model and the virtual workstation has been compared with the real one in terms of both plant layout and work method. In this regard some wrong working postures, wrong motions and redundant motions have been corrected or deleted.

At the end of the validation phase, the authors concluded that the simulation model is capable of recreating the real assembly workstation with satisfactory accuracy.

4. THE ERGONOMIC STANDARDS

The main goal of the paper is to achieve the ergonomic effective redesign of the most critical workstation (assembly workstation) of a manufacturing plant by means of an integration between M&S tools and several ergonomic standards. In this regards after the simulation model development, next step was to evaluate the Assembly workstation in terms of ergonomic issues. In effect, eMWorkplace provides the user with several ergonomic indexes based on the most known ergonomic standards (Burandt Schultetus, NIOSH 81, NIOSH 91, OWAS and Garg analysis). The examination of such indexes has allowed the authors to identify all the ergonomic problems related to the workstation. Before getting into the details of the ergonomic analysis, let us describe the ergonomic indexes and the relative ergonomic standards used in the simulation study.

The ergonomic indexes are the lift indexes (evaluated by Lift analysis), the stress levels associated to working postures (evaluated by using the OWAS analysis) and the energy expenditure associated to each activity (evaluated by using the Garg analysis).

The lift analysis consists of three ergonomic methods used for the measurement of the effort that affects the elevation of a weight and its transportation towards another point. The methods are: Burandt Schultetus, NIOSH81 and NIOSH91.

The Burandt-Schultetus analysis detects the maximum weight that a working person can lift (maximum permissible force).

The maximum permissible force can be evaluated by using equation (1):

$$PF = G * C * AJ * RF \quad (1)$$

where,
G is a coefficient for the worker's gender;
C is a coefficient for the worker's health condition;
AJ is a coefficient for worker's age and type of job;
RF is the reference force.

Note that the AJ (Age and Job factor) depends on the effort type (i.e. static or dynamic), the worker's age, the shift time (i.e. 8 hours) and the effort frequency. The RF takes into consideration the torso weight movement, the hands use (i.e. one or two hands), the number of persons performing the operation (i.e. one or two persons), the effect of secondary jobs and the maximum force. In turns, the torso weight movement depends on the lower and upper grasp height and motion frequency; the maximum force depends on body size class (anthropometric measure), upper grasp height and distance of grasp from the body.

Moreover, the method requires several input parameters regarding the physical conditions, age and gender of the worker, the load weight, the lifting frequency (measured in lifts per minute) and the total task duration.

The maximum permissible force is then compared to the current actual force (AF) being exerted. Three different cases can be distinguished:

- Case 1: the maximum permissible force do not exceeds the actual force then an ergonomic intervention is required;
- Case 2: the maximum permissible force is equal to the actual force, then a corrective intervention is necessary in the near future;
- Case 3: the actual force is lower than the maximum permissible force, then no ergonomic intervention is required.

The NIOSH 81 method calculates the action limit (AL) and the maximum permissible limit (MPL). AL is the weight value which is permissible for 75% of all female and 99% of all male workers. MPL is the weight value which is permissible for only 1% of all female and 25% of all male workers. Three different cases can be distinguished:

- Case A: the action limit exceeds the maximum permissible limit then an ergonomic intervention is required;
- Case B: the action limit is equal to the maximum permissible limit, then a corrective intervention is necessary in the near future;
- Case C: the actual limit is lower than the maximum permissible limit, then no ergonomic intervention is required.

The NIOSH 91 analysis, additionally to the NIOSH 81 method, includes the recommended weight limit (RWL) and the lifting index (LI). The RWL is the load that nearly all healthy workers can perform over a substantial period of time for a specific set of task conditions. The LI is calculated as ratio between the real object weight and the recommended weight limit. Three different cases can be distinguished:

- If LI value is less than 1 the lifting task is not hazardous for some of the population;
- If the LI value is equal to 1 the lifting task could be hazardous for some population;
- If the LI value is greater than 1 the lifting task is hazardous for some of the population.

The LI is the approximate value of the relative level of strain.

Note that NIOSH 81 and NIOSH 91 require as input parameters data regarding worker postures at the origin and destination of the lift, object coupling and the duration of the specified task.

As concerns the OWAS analysis, it carries out a qualitative analysis of the worker's movements during a working process. The analysis calculates the stress associated to each body posture and classifies them in one of the following four stress categories:

- Category 1: the stress level is optimum, no corrective interventions are required;
- Category 2: the stress level is almost acceptable, corrective interventions are necessary in the near future;
- Category 3: the stress level is high, corrective interventions are required as soon as possible;
- Category 4: the stress level is very high, corrective interventions must be carried out immediately.

Finally, the Garg analysis calculates the total amount of energy spent during the manual operations. The analysis splits up a specified operation into smaller steps calculating for each of them the Energy Expenditure (EE); the sum of these separate steps represents the total Energy Expenditure for the activity.

As input parameters, such analysis requires information concerning load weight and body weight as well as gender of the working person.

5. SIMULATION RESULTS

In this section the authors discuss the ergonomic analysis results as output of a simulation model run.

The OWAS method was the first one applied in this research work. As previously stated, OWAS is a method that classifies each body posture within one of four stress categories (see section 4). The main advantage of using a simulation tool, as eMWorkplace, is that, during the simulation, the postures are analyzed in real-time and the effects are shown visually. As a result, the program points out which body parts are most affected (figure 3).

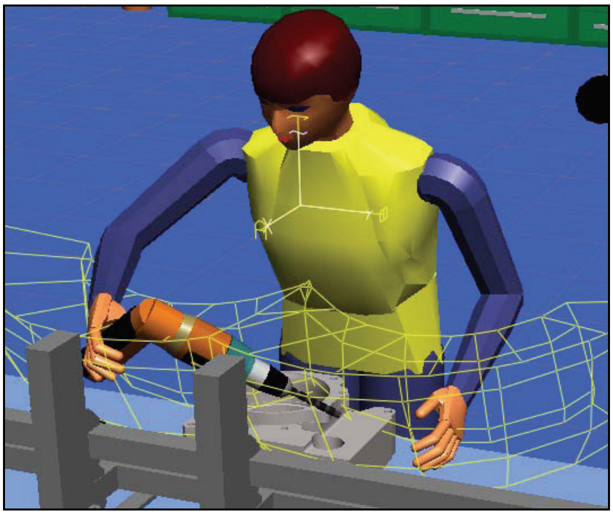
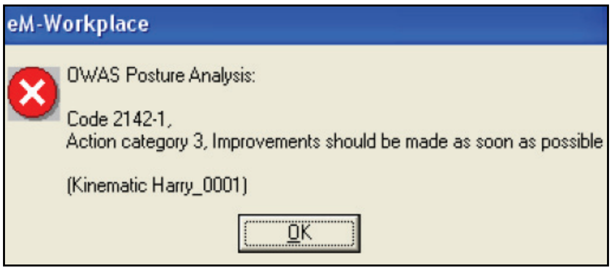


Fig. 6: Results of the OWAS analysis

Moreover as soon as the software identifies a harmful working posture, a message window appears reporting the category it belongs to. Figure 7 shows a message window for a working posture belonging to the Category 3.



OWAS Posture Analysis:

Code 2142-1,
Action category 3, Improvements should be made as soon as possible

(Kinematic Harry_0001)

Figure 7: Message window for Category 3 working posture

When the OWAS method is applied to the Assembly workstation the program assigns a category 3 to the task of taking manually the mechanical component from the pallet. Note that category 3 indicates a high level of effort that produces negative effects on the worker's health, so corrective interventions are required as soon as possible. The color orange, that represents category 3, is visible on the back and the legs of the worker, which are the most affected parts in the movement (figure 8).

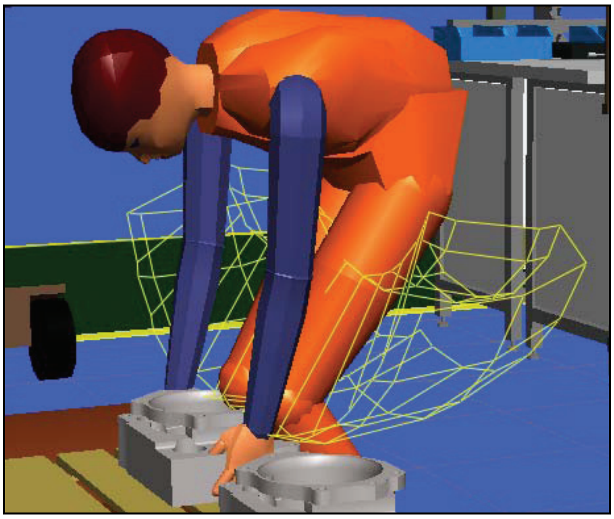


Figure 8: Example of a body posture belonging to the category 3

When the operator uses the working tools (hand drill, hexagonal wrench, hammer) for performing the assembly activities, the software indicates a category 2. A category 2 body posture can have adverse effects on the muscular system so that corrective interventions are necessary in the near future. In this case the most affected body part is the back (note that the color yellow is associated to a body part belonging to the category 2).

Finally, the software reports another Category 2 as soon as the worker gets manually the assembled workpiece, whose weight is 23 Kg, and puts it near the mechanical arm location.

As the next step, the ergonomic process is studied through the Lift Analysis.

The Lift Analysis have been used for measuring the strain resulting from human lifting and carrying activities. In this regards, as previously stated in the "ERGONOMIC STANDARDS" section, three ergonomic methods have been carried out: Burandt Schultetus, NIOSh 81 and NIOSH 91.

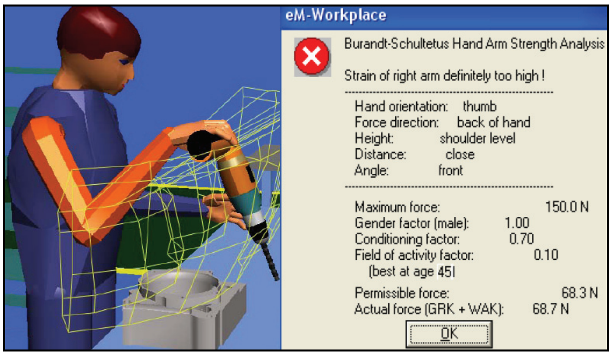
The first one being applied to the Assembly Workstation was the Burandt Schultetus method. It calculates the maximum permissible force a worker can lift and compare such limit to the actual force being exerted. The input parameters were introduced based on the working cycle, for normal grabbing of the objects and a total duration of the task based on the average time of a Shop Order processing. Note that a typical Shop Order requires the production of 25 assembled workpiece.

Table 3 reports the input parameters inserted into the simulation software for correctly carrying out the Burandt Schultetus analysis.

Table 3: Burandt Schultetus input parameters

Worker physical characteristics	
Physical Condition	normal
Age	45
gender	male
Objects being moved Load weight (Kg)	
mechanical component	14
flange	9
hand drill	6
hexagonal wrench	0,2
hammer	0,6
Total task duration (25 workpiece)	
Time (sec)	6125

As soon as high stress levels occur, a message window appears reporting the maximum permissible and actual force values. Moreover the most affected body parts appear orange colored (figure 9).



```
Burandt-Schultetus Hand Arm Strength Analysis

Strain of right arm definitely too high !
------------------------------------------------
  Hand orientation:   thumb
  Force direction:    back of hand
  Height:             shoulder level
  Distance:           close
  Angle:              front
------------------------------------------------
  Maximum force:                          150.0 N
  Gender factor (male):          1.00
  Conditioning factor:           0.70
  Field of activity factor:           0.10
     (best at age 45)
  Permissible force:                      68.3 N
  Actual force (GRK + WAK):        68.7 N
```

Figure 9: Results of the Burandt Schultetus analysis

When the Burandt Schultetus method is applied to the Assembly workstation the program detects three critical lifting operations: the grabbing of the mechanical component from the pallet (1), the handling of the hand drill (2) and the carrying of the assembled workpiece to the mechanical arm location (3); in effect each of them is characterized by a PF value lower than the AC being exerted. Table 4 reports for each operation the permissible force and actual force values.

Table 4: Burandt Schultetus returned values

Operation	**Maximum permissible force (N)**	**Actual force (N)**
(1)	67,8	68,7
(2)	68,3	68,7
(3)	52,3	112,8

As concerns the operations (1) and (2) the PF values are very close to the AF values; in fact in both cases the AF exceeds the PF no more than 1 N. It means that the stress level is almost acceptable and corrective interventions are necessary in the near future.

On the other hand, the operation (3) has extremely adverse effects on the muscular system; in fact a huge gap between the PF and the AF (the gap value is about 60 N) characterizes it. In this regards, corrective interventions must be carried out immediately.

NIOSH 81 and NIOSH 91 methods complete the Lift analysis. NIOSH 81 calculates the AL and the MPL. NIOSH 91 calculates the RWL and the LI. As input parameters, both methods requires information regarding worker posture at the origin and destination of a lift, object coupling (i.e. grip quality) and the duration of the specified task.

Worker posture at the origin and destination of a lift were identified during the program code writing. In fact, eMWorkplace provides the user with specific commands for denoting the beginning and the end of each lifting operation (in turn, the initial and final worker postures). In this regards the commands “LIFT BEGIN” and “LIFT END” were accomplished respectively just before and after the code lines written for performing the lifting operation.

Grip quality was set as normal for each object being lifted.

Task duration was based on the average time for processing a typical Shop Order made by 25 assembled workpieces; the average time is about 6125 seconds.

NIOSH 81 and NIOSH 91 methods were carried out for each lifting operation performed during the assembly process. As Burandt Schultetus analysis, the grabbing of the mechanical component from the pallet (1), the handling of the hand drill (2) and the carrying of the assembled workpiece to the mechanical arm location (3) were identified as critical movements for the operator.

Table 5 and table 6 show respectively the NIOSH 81 and NIOSH 91 results for the critical operations.

Table 5: Results of NIOSH 81 analysis

NIOSH 81			
Operation	**MPL (Kg)**	**AL (Kg)**	**Object weight (Kg)**
(1)	31,45	10,48	14
(2)	26,54	8,96	6
(3)	27,58	9,19	23

Table 6: Results of NIOSH 91 analysis

NIOSH 91				
Operation	**RWL (Kg)**		**Object weight (Kg)**	**LI**
(1)	**Origin**	8,83	14	1,58
	Destination	11,30	14	1,23
(2)	**Origin**	6,23	6	0,96
	Destination	6,11	6	0,98
(3)	**Origin**	7,26	23	3,16
	Destination	6,15	23	3,73

Let us describe the NIOSH 81 results. The MPL values exceed the object weight for each lifting operation. It means that the lifting operations no have adverse effects on the muscular system and, in turn, no corrective interventions are necessary. However, note that the MPL represents the weight value which is permissible for only 1% of all female and 25% of all male workers. We can conclude it is not a significant index due to the little part of the population it takes into consideration. On the other hand, the weight value which is permissible for 75% of all female and 99% of all male workers (action limit) is lower than the objects weight being lifted.

As concerns the operation (2), the AL value is very close to the object weight (the gap is 0,04 Kg) so that the stress level is almost acceptable; in this case corrective interventions are needed in the near future. On the contrary, AL is lower 3,52 Kg and 13,81 Kg than the objects weight respectively for operation (1) and operation (3). In the first case the stress level is high and corrective interventions are necessary as soon as possible, in the second case the stress level is very high and corrective interventions are immediately required.

Concerning the description of NIOSH 91 results, let us consider the LI values. As previously stated, it is calculated as ratio between the object weight and the RWL value.

Operation (1) and operation (3) are characterized by LI values higher than one in both the origin and the destination points of the lifting operations. In the first case the LI values (1,58 and 1,23) suggest corrective interventions as soon as possible. In the second case the very high LI values (3,16 and 3,73) require corrective interventions immediately.

Considering the operation (2), 0,96 and 0,98 represent acceptable LI values so that no corrective interventions are needed.

Finally, the ergonomic study is completed by the Garg analysis. It calculates the total amount of energy spent during the manual operations. Note that such value provides information on the stress rate of the activity. As input parameters it requires information regarding worker physical characteristics (physical condition, age and gender) and the weight of objects being moved (objects weight is reported in table 3). At the end of a simulation run, the total amount of energy spent during the whole working shift is about 1437 Kcal.

Let us summarize the most noteworthy aspects provided by the ergonomic analysis.

The ergonomic process has pointed out the high level of ergonomic risks affecting the Assembly workstation. In particular three operations were identified as the most harmful for the operators muscular system.

The grabbing of the mechanical component from the pallet (1) hurts the back and the legs of the worker.

The handling of the hand drill (2) affects the operator right hand-arm system.

Finally, the carrying of the assembled workpiece to the mechanical arm location (3) causes pain to the worker back and arms.

We can conclude the ergonomic effective design of the workstation is required for preserving the workers health. Note that the ergonomic effective design should also aim at reducing the total amount of energy spent during the assembly process.

Section 6 describes the design guidelines for developing an improved workstation configuration in terms of ergonomic issues.

6. THE ERGONOMIC WORKSTATION DESIGN

Once the initial situation was studied, a new workstation configuration was developed for eliminating the movements that could cause worker injuries.

Let us list the critical operations affecting the workstation and describe for each of them the solution the authors propose for preventing the ergonomic problems.

- the grabbing of the mechanical component from the pallet: it requires to the worker continuous bending due to the location of the pallet on a hand cart high 20 cm. Operator back and legs are the most affected body parts. The authors decided to substitute the initial hand cart for an adjustable one. This change allows to custom the pallet position in height according to the operators needs. Note that the adjustable hand cart will be used also for moving the assembled workpiece during the assembly process. Figure 10 and figure 11 show respectively the actual configuration and final solution.

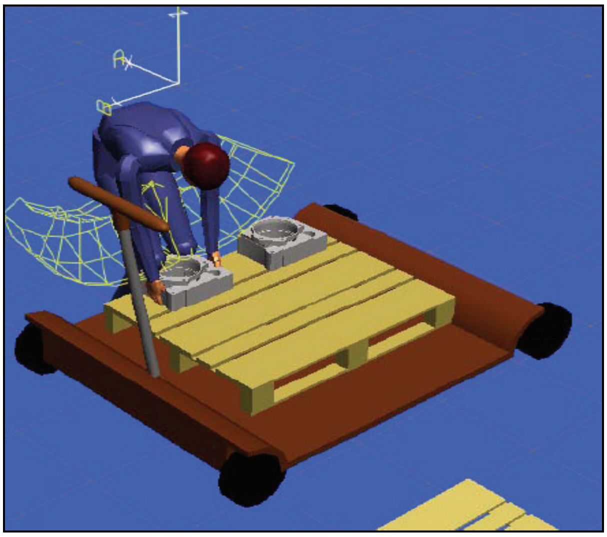


Figure 10: Actual workstation configuration for the grabbing the mechanical component from the pallet

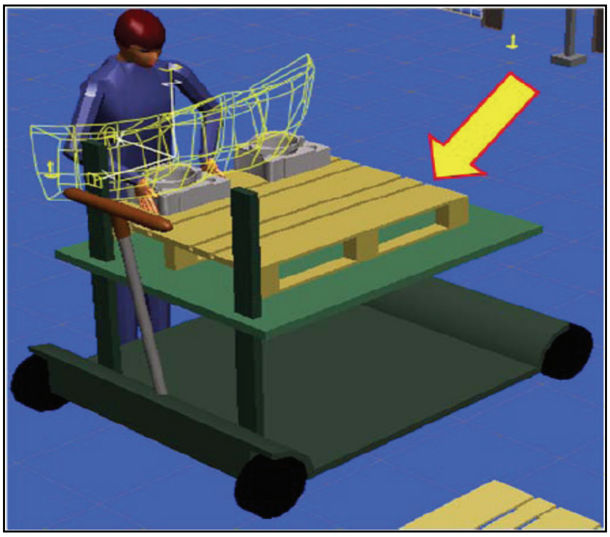


Figure 11: Final workstation configuration for the grabbing the mechanical component from the pallet

- the handling of the hand drill: it causes pain to the operator right hand-arm system. As previous stated, the hand drill is used for drilling the mechanical component 5 times. In a first moment the authors thought to modify the mechanical component and the hand drill positions for making them easier to manage. Actually, this operation is strongly affected by the hand drill weight (6 Kg), so any change regarding the objects position would have been a useless solution. In this regards, the authors advised the company top management to purchase a lighter hand drill. In particular they proposed a 1.6 Kg hand drill characterized by an ergonomic handle.
- the carrying of the assembled workpiece to the mechanical arm location: 670 cm must be walked carrying manually the assembled workpiece, whose weight is about 23 Kg. Obviously, the worker back and arms are the most stressed body parts. As previous stated, the authors proposes to adopt the adjustable hand cart for performing such operation. In this way, the worker has only to place the assembled workpiece on the hand cart and then push it to the final destination. Figure 12 and figure 13 show respectively the actual configuration and the final solution.

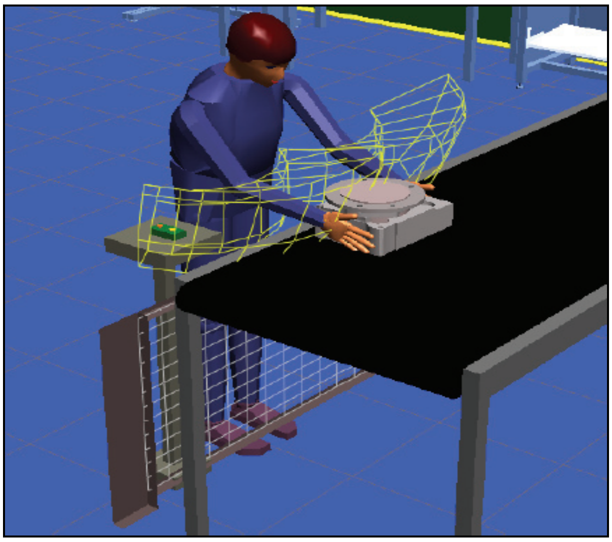


Figure 12: Actual workstation configuration for the carrying the assembled work piece to the mechanical arm location

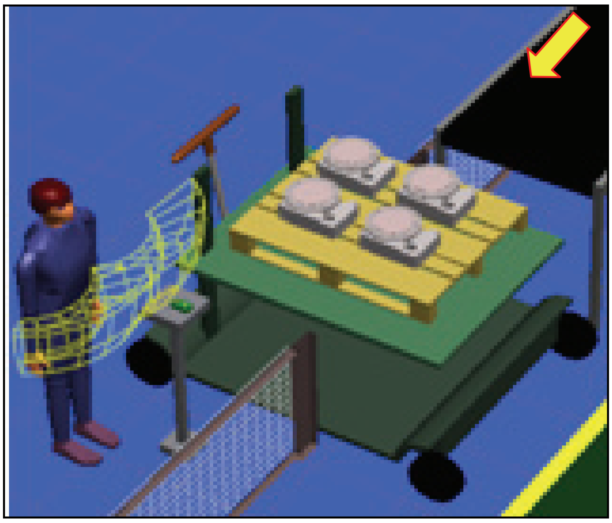


Figure 13: Final workstation configuration for the carrying the assembled work piece to the mechanical arm location

Figure 14 shows the final workstation configuration. Two red boxes point out the workstation changes respect to the initial configuration.

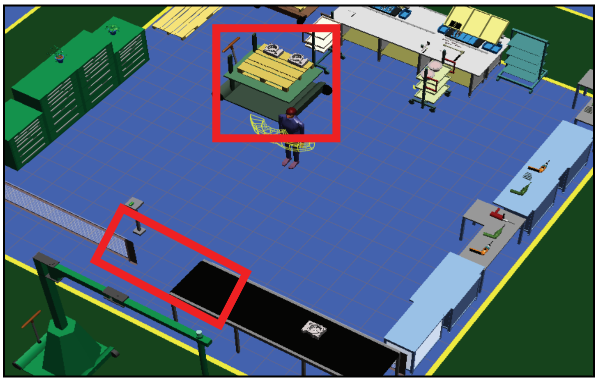


Figure 12: Final Assembly workstation

The use of the simulation allows to check the goodness of the changes before they are implemented in the real system. To this end, the final workstation configuration has been tested by means of the simulation model. In this regards the ergonomic analysis have been carried out for evaluating the ergonomic risks inside the proposed workstation configuratiom. The simulation results point out the ergonomic effective design of the workstation. In particular the ergonomic issues related to the three critical operations have been solved and no other ergonomic problems have been detected.

The OWAS analysis does not reveal any particular posture problem. The stress level related to each body posture is optimum.

According to the Lift analysis (Burandt Schiltetus, NIOSH 81 and NIOSH 91) no lifting problems affect the workstation.

Consider the Garg analysis. The total amount of energy spent during the whole working shift is about 1203 Kcal. It means that the EE reduction is about 17% respect to the initial situation.

In conclusion, the goodness of the configuration is also sustained by the low economic efforts required for the workstation changes implementation in the real system. In effect the company has only to incur the costs related to the purchase of the hand drill and the adjustable hand cart. Table 7 reports the objects costs.

Table 7: Objects costs

Object	**Costs ($)**
Adjustable hand cart	594.40
Hand drill	60.00
Total	654.40

The workstation relevant ergonomic improvements completely justified the total amount of 654.40 $.

7. CONCLUSION

The main goal of the paper is to achieve the ergonomic effective redesign of the most critical workstation (assembly workstation) of a manufacturing plant by means of an integration between M&S tools and several ergonomic standards. The authors started the research work by modelling the actual configuration of the workstation. The simulation model has been developed by using the CAD software Pro-Engineer and the simulation software eMWorkplace. After the simulation model validation, several egonomic analysis were accomplished for identifying the ergonomic problems related to the actual workstation. A new workstation configuration was developed for eliminating the movements that could cause worker injuries.

The final configuration of the workstation has acceptable ergonomic risks (related to lifting activities and working postures) and guarantees a smaller amount of energy required for performing all the operations.

Further researches are still on going (in cooperation with the same company) for analyzing the remaining workstations of the Production area.

REFERENCE

Banks. J., 1998. *Principles of Simulation, Handbook of Simulation*. New York: Wiley Interscience.

Bell, P.C., O'Keefe, R.M., 1994. Visual interactive simulation: a methodological perspective. *Annals of Operations Research*, 53, 321-342.

Bocca, E., Longo, F., 2008. *Simulation Tools, Ergonomics Principles and Work Measurement Techniques for Workstations Design*. Proceedings of Summer Computer Simulation Conference, 15-18 June, Edinburgh (UK).

Carrasco, C., Coleman, N., Healey, S., 1995. Packing products for customers, An ergonomics evaluation of three supermarkets checkouts. *Applied Ergonomics*, 26, 101-108.

Chang, S.-W., Wang, M.-J. J., 2007. Digital Human Modeling and Workplace Evaluation: Using an Automobile Assembly Task as an Example. *Human Factors and Ergonomics in Manufacturing,* 17, 445-445.

Das, B., Sengupta, A. K., 1996. Industrial workstation design: a systematic ergonomics approach. *Applied Ergonomics*, 27, 157-163.

De Sensi, G., Longo, F., Mirabelli, G., 2007-a. *Modeling & Simulation Based Approach for Optimizing Seal Press Workstation in a Manufacturing System*, Proceedings of Business and Industry Symposium, March 25-29, USA.

De Sensi, G., Longo. F., Mirabelli, G., 2007-b. *Ergonomic work methods optimization in a three dimensional environment*, Proceedings of Summer Computer Simulation Conference, July 15-18, San Diego, California, USA

Engström, T., Medbo, P., 1997. Data collection and analysis of manual work using video recording and personal computer techniques. *International Journal of Industrial Ergonomics,* 19, 291-298.

Feyen, R., Liu, Y., Cha, D., Jimmerson, G., Joseph, B., 2000. Computer-aided ergonomics: a case study of incorporating ergonomics analyses into workplace design. *Applied Ergonomics,* 31, 291-300.

Garg A., 1976. *A metabolic rate prediction for manual materials handling jobs*. Dissertation, University of Michigan.

Grant, K.A., Habes, D.J., Bertsche, P.K., 1997. Lifting hazards at a cabinet manufacturing company: Evaluation and recommended controls. *Applied Occupational and Environmental Hygiene*, 12, 253-258.

Jones, T., Strickfaden, M., Kumar, S., 2005. Physical demands analysis of occupational tasks in neighborhood pubs. *Applied Ergonomics,* 36, 535-545.

Kadefors, R., Forsman, M., 2000. Ergonomic evaluation of complex work: a participative approach employing video computer interaction, exemplified in a study of order picking. *International Journal of Industrial Ergonomics,* 25, 435-445.

Kharu, O., Harkonen, R., Sorvali, P., Vepsalainen, P., 1981. Observing working postures in industry: Examples of OWAS application. *Applied Ergonomics*, 12, 13-17.

Lin, R.T., Chan, C.-C., 2007. Effectiveness of workstation design on reducing musculoskeletal risk factors and symptoms among semiconductor fabrication room workers. *International Journal of Industrial Ergonomics,* 37, 35-42.

Longo, F., Mirabelli, G., Papoff, E., 2006. Effective Design of an Assembly Line Using Modeling & Simulation. *Proceedings of the Winter Simulation Conference*, Monterey, California, USA.

Longo, F., Mirabelli, G., Papoff, E., 2005. *Techiche di analisi avanzate per la progettazione efficiente delle postazioni di assemblaggio manuale*, SdA – Soluzioni di Assemblaggio, VNU Business Publications Italia.

Longo, F., Mirabelli, G., Papoff, E., 2006-a. *Material Flow Analysis and Plant Lay-Out Optimization of a Manufacturing System*, International Journal of Computing, 5(1), 107-116.

McAtamney, L., Corlett, E.N., 1993. RULA: a survey method for the investigation of work-related upper limb disorders. *Applied Ergonomics*, 24, 91-99.

Mital, A., Ramakrishnan, A., 1999. A comparison of literature-based design recommendations and experimental capability data for a complex manual materials handling activity. *International Journal of Industrial Ergonomics,* 24, 73-80.

Neumann, W.P., Wells, R.P., Norman, R.W., Kerr, M.S., Frank, J., Shannon, H.S., OUBPS Working Group, 2001. Trunk posture: reliability, accuracy, and risk estimates for low back pain from a video based assessment method. *International Journal of Industrial Ergonomics,* 28, 355-365.

Niosh Technical Report 81-122. National Institute for Occupational Safety and Health (Hrsg.). Work practices guide for manual lifting. *Center for Disease Control, U.S. Department of health and human services*, Cincinnati, OH, USA: NTIS 1981.

Russell S. J., Winnemuller L., Camp J. E., Johnson P. W., 2007. Comparing the results of five lifting analysis tools. *Applied Ergonomics*, 38, 91-97.

Russell S. J., Winnemuller L., Camp J. E., Johnson P. W., 2007. Comparing the results of five lifting analysis tools. *Applied Ergonomics*, 38, 91-97.

Santos, J., Sarriegi, J. M., Serrano, N., Torres, J. M., 2007. Using ergonomic software in non-repetitive manufacturing processes: A case study. *International Journal of Industrial Ergonomics,* 37, 267-275.

Sargent, R.G., 1996. Verifying and validating simulation models. *Proceedings of the Winter Simulation Conference*. pp. 55-64. Piscataway (New York, USA).

Schultetus, W., 1980. Daten, hinweise und beispiele zur ergonomischen arbeitsgestaltung. Montagegestaltung, Verlag TÜV Rheinland Gmbh, Köln.

Scientific Support Documentation for the Revised 1991 NIOSH Lifting Equation: Technical Contract Reports, May 8, 1991, NTIS No. PB-91-226-274.

Scott, G. B., Lambe, N. R., 1996. Working practices in a pherchery system, using the OVAKO Working posture analysing System (OWAS). *Appied Ergonomics,* 27, 281-284.

Temple, R., Adams, T., 2000. Ergonomic Analysis of a Multi-Task Industrial Lifting Station Using the NIOSH Method. *The Journal of Industrial Technology*, 16, 1-6.

Van Wendel de Joode, B., Burdorf, A., Verspuy, C., 1997. Physical load in ship maintenance: Hazard evaluation by means of a workplace survey. *Applied Ergonomics,* 28, 213-219.

Vedder, J., Hellweg, 1998. Identifying postural hazards with a video-based occurrence sampling method. *International Journal of Industrial Ergonomics*, 22, 4-5.

Waters, T.R., Vern, P.A., Garg, A., 1994. Application Manuals for the Revised NIOSH Lifting Equation. *U.S. Department of health and human services, National Institute for Occupational Safety and Health*, Cincinnati, OH, USA.

Wright E.J., Haslam R.A., 1999. Manual handling risks and controls in a soft drinks distribution centre. *Applied Ergonomics*, 30 , 311-318.

AUTHORS BIOGRAPHY

ANTONIO CIMINO was born in Catanzaro (Italy) in October the 1th, 1983. He took his degree in Management Engineering, summa cum Laude, in September 2007 from the University of Calabria. He is currently PhD student at the Mechanical Department of University of Calabria. His research activities concern the integration of ergonomic standards, work measurement techniques, artificial intelligence techniques and Modeling & Simulation tools for the effective workplace design. He collaborates with the Industrial Engineering Section of the University of Calabria to research projects for supporting innovation technology in SMEs.

GIOVANNI MIRABELLI was born in Rende in 1963 and he took the degree in Industrial Engineering at the University of Calabria. He is currently researcher at the Mechanical Department of University of Calabria. His research interests include ergonomics, methods and time measurement in manufacturing systems, production systems maintenance and reliability, quality. He has published several scientific papers participating as speaker to international and national conferences. He is actively involved in different research projects with Italian and foreign universities as well as with Italian small and medium enterprises.

OPTIMIZING TIME PERFORMANCE IN REACHABILITY TREE- BASED SIMULATION

Miguel Mujica[(a)], **Miquel Angel Piera**[(b)] , **Mercedes Narciso**[(c)]

Autonomous University of Barcelona, Department of Telecommunications and Systems Engineering.
Barcelona, Spain.

[(a)]MiguelAntonio.Mujica@uab.es, [(b)]MiquelAngel.Piera@uab.cat, [(c)]Mercedes.Narciso@uab.es,

ABSTRACT

This paper presents the highlights of a two-step algorithm based on the reachability tree for optimizing industrial system models coded with the Coloured Petri Net formalism. The first step of the algorithm consists in generating all the possible states that can be reached by the system, and the second step of the algorithm focuses on updating the time stamp values of the most promising state subspaces of the system. Some tests with benchmarking problems have been made to evaluate the efficiency of the second part of the algorithm, and the results obtained are analyzed in order to identify the algorithm processes that need improvements. Based on the results, an heuristic has been developed to improve the performance with a better choice of states to be expanded during the generation of the reachability tree.

Keywords: Coloured Petri nets, reachability tree, simulation, decision support systems, optimization time.

1. INTRODUCTION

Due to changes in the market rules (i.e. changes from high volume production to high diversity), during the last decade, companies shifted their methods for producing goods to more flexible schemas of production, distribution and logistics. With this transformation, companies expected to improve their productivity as more flexibility was gained by its value chain. Soon it became clear that the more flexible the processes were, the more difficult became it to coordinate all the elements of the chain (suppliers, production elements, distribution entities, etc).

Therefore, it became important to tune decision support tools that could help decision makers to coordinate in the most efficient way the different processes and elements of the system.

Several approaches from diverse knowledge areas have been developed to face some aspects of these problems. In later years, with the improvement of digital computers, digital simulation has emerged as a tool capable to give support in the decision making process of industrial systems. Petri Nets (PN) is a modeling formalism that has been applied successfully to develop accurate discrete event system models of industrial and logistic systems. It presents certain modeling characteristics which make it appropriate to be applied in these fields; in particular it supports models for concurrency, parallelism and conflicting situations. Coloured Petri Nets (CPN) is a high abstraction level modeling formalism that allows modeling very complex industrial systems in a more simplified representation (which result easier to maintain) than the model that could be obtained with Petri Nets (Jensen 1997). Both CPN and PN have been used for modeling the structural behavior of systems but they can also be employed to analyze system performance. In order to achieve this objective the formalism has been extended with time representations (time stamps, global clock) (Jensen 1997).

The reachability tree is a quantitative analysis tool used in PN and CPN which allows storing all the different states of a system in a tree-based structure. In general it is employed to verify Petri nets behavior through the exploration of the state space (Jensen, Christensen, Hard, Holzman). In the work presented in this article, the reachability tree approach has been used to develop a two step efficient algorithm with the objective to improve industrial systems performance.

1.1. The Reachability Tree

The reachability tree is a directed graph in which the nodes represent the set of reachable states of the system and the arcs correspond to the state changes. Such a graph represents all the possible system’s scenarios and can be used to verify and analyze some properties of the system, but it shows some disadvantages when it is used for optimizing purposes:

- In order to detect previously appeared states, all the generated nodes must be stored in computer memory causing most of the times memory saturation due to the state explosion problem.
- The simulation performance is reduced as the memory burden grows when the storage of states takes place.
- In order to simulate the system some nodes need to be reevaluated, increasing the computational time.

- When evaluating old nodes time characteristics, an extra computational effort appears because some branches must be updated according to the combination of time stamps.

There have been some attempts to overcome the problem of memory saturation due to the exponential growth when new states are generated. Jard (1991) and Christensen (2002) developed respectively an approach that does not store all the states of the model and throws away states on the fly, minimizing the amount of computer memory used to explore the state space.

These approaches have the advantage that reduce the run-time of the state space exploration and result very efficient for analyzing and verifying system properties such as dead locks and safety properties, but the old node analysis made is not as efficient as it could be done with a different approach.

Holzmann (1998) developed an algorithm that is able to maintain all the states based in the hashing principle. This approach uses an efficient way of storing the different states of the system; however a main disadvantage appears when two different states are mapped to the same hash value (collision).

The state analysis that has been implemented in the algorithm presented in this article was based on the principles reported by Narciso and Piera (2005). It stores all state space information and evaluates only the most promising branches in the tree. This approach which does not overcome completely the problems of speed and memory, is capable to generate results analyzing the most promising branches of the tree and updating the time stamp values based on the decision rules reported in the same work. The state analysis has been combined with the fast search and efficient transition evaluation algorithms developed by Mujica and Piera (2006, 2007) resulting in an efficient tool for exploring the state space with optimization purposes, and a very promising one to solve real life industrial problems.

2. REACHABILITY TREE-BASED SIMULATOR

A simulator to optimize industrial system performance based on time analysis of old nodes stored in the reachability tree has been coded.

2.1. Time representation for the Reachability Tree

To represent time in the reachability tree, it is necessary to attach a global clock to each state besides the correspondent time stamps of the tokens that belong to the marking. To understand how time flows between states when using the time extension it is proposed to take any node in the state space (one suggestion is to take the root node), and follow one of the directed arcs to its successors (child nodes); the difference between the clocks represents the amount of time that has past when going from one node to the next one, i.e. the time that takes the system to change from one state to the other. If the same process is repeated with the subsequent nodes until getting the objective state it will be possible to obtain the complete flow time for the process. A representation of a reachability tree with its global clocks is presented in Figure 1.

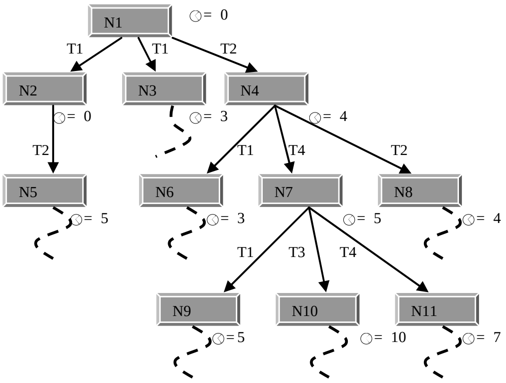


Figure 1: Time Extension for Reachability Tree

In this example the reachability tree is generated under the CPN rules (color enabling rules). Time advances during the generation of children nodes based on the rule that the global clock must take the minimal time between all time stamps of the enabling tokens (Narciso and Piera 2005). Using these rules to explore the reachability tree, it will be possible to analyze the state evolution behavior using time stamps and the global clock.

2.2. Updating Time in the State Space

After exploring the reachability tree, all the possible system states are generated (i.e. a bounded state space) and stored in the tree structure. The old nodes are stored altogether with their time characteristics in a separate list for later analysis.

When time is involved in the analysis, it is common to find old nodes which could differ one from the other in their time extension characteristics (time stamps and global clock). Since the same state can be reached from a different node, the time extension characteristics of old nodes should be properly updated in order to deal with the timed state space. However when dealing with optimization problems instead of system verification properties, it is possible to save computational time avoiding time evaluation for the old nodes which will not lead to the solution of the optimization problem.

The optimization of the paths in the state space is carried out making a time analysis for the old nodes. The algorithm compares the time stamps and global clock, and updates the time values of the best path that leads to the objective state. Narciso and Piera (2005) proposed an approach which analyses the old nodes in a way that the algorithm avoids the complete exploration of branches in the reachability tree. It updates only the ones that improve the paths which lead to the objective states (optimization). The authors proposed a rule for deciding between two old node states basing the decision only on the time stamps of the markings. One example of how to use the mentioned rule is shown in Figure 2.

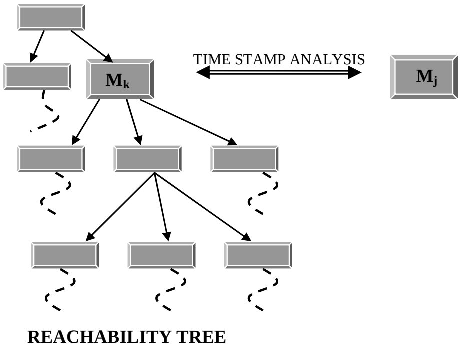


Figure 2: The time analysis of an Old Node

There are two markings representing the same state. One state M_k which has been already stored during the generation of the reachability tree is compared with another marking M_j which represents the same state but it has different time stamps values. In this case three possible results can be obtained from the evaluation:

- The time stamp values of the marking M_j are less than the time stamp values of the original marking M_k. Therefore it can be concluded that the marking M_j will produce states with better time values than the ones that are actually stored in the reachability tree. In this case the original marking M_k is replaced by the marking M_j, and the time values are updated for the best branch that goes from the original marking to the final one.
- Dealing with time minimization, the time stamp values of the marking M_j are greater than the values of the marking M_k. In this case it can be concluded that the M_k marking is better than the marking M_j and the original marking M_k will be kept without making any value updating.
- Some time stamp values in the marking M_j are better than the correspondent ones in the marking M_k and some are worse. For this outcome nothing can be concluded about the two markings. In this case a depth search must be done using the M_j time stamps values in order to get to the final state and then a decision based on the final time must be taken. If the final time resulting from using the M_j time values is better than the stored one, the marking M_j will replace the original marking M_k and the time stamps of the branch that goes from M_j to the final state.

2.3. A Two- Step Reachability Tree Algorithm

In the first step of the algorithm, the CPN rules are used to generate the complete state space. All the possible states of the system are generated together with their time values which depend mainly on the exploring sequence of the algorithm. The repeated states that appear during the exploration (old nodes) are stored in a separate list with their correspondent time stamps and global clock for a later analysis.

In the second step of the algorithm (when all the states of the system have been already generated) the stored old nodes are analyzed comparing their time stamp values with the correspondent values in the nodes stored in the reachability tree. This procedure is carried out with the purpose of finding the time values that reduce the most the final time of the sequence of states that go from the initial state to the final one. Therefore the time span for the whole procedure is optimized. The old node analysis is done following a simple order in the old node list (from the first to the last element).

The time analysis of tokens is based on the algorithm and updating rules proposed by Narciso and Piera (2005) which avoid the exhaustive analysis of all branches at lower levels in the state space when a better node is selected and the time stamps of the offspring nodes are updated.
The two steps of the algorithm are outlined in Figure 3

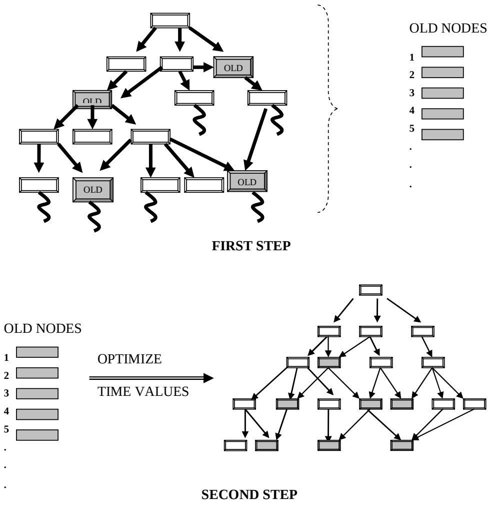


Figure 3: The Two-Step Algorithm.

In the first step the shaded nodes correspond to the old nodes that appear during the state generation, and are stored in the old node list. In the second step of the algorithm the list is evaluated focusing on the time stamp values of the nodes comparing them with the ones stored in the reachability tree, applying the evaluation algorithm previously mentioned.

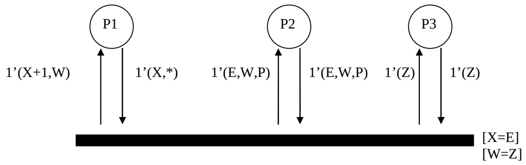
3. TESTING THE SIMULATOR/OPTIMIZER

The simulator performance was tested by modeling some job-shops which appear to be well-known benchmarks (Dauzére et al 1994). Such job-shops are the 3x3, 5x5 and the 6x6 job-shops. The objective of the evaluation was to establish the efficiency of the second step of the algorithm throughout the comparison of time stamps between repeated states when the depth exploration is done.

3.1. The Job-Shops

The job-shop in its different modalities is a benchmarking problem which consists in a certain number of jobs that must follow a specified sequence of tasks through different machines. In the 5x5 job-shop five jobs must go through processes in five different machines, the goal of these benchmarks is to obtain the sequence that minimizes the time span of the whole procedure. The different job-shop models were coded with the CPN formalism.

A typical model for a job-shop is presented in figure 4.



COLOUR	DEFINITION	DESCRIPTION
J	Int 11..33	Job and Task Identifier
W	Int 1..3	Machine Needed for the Next Job
X	Int 11..33	Job and Task in progress
E	Int 11..33	Job and Task Identifier
P	INTEGER	Time Spent for each Job
Z	Int. 1..3	Available Machines

Figure 4: The CPN Model of the 3x3 Job-Shop

The place node P1 stores the tokens which represent the information of the job and task numbers (color J) and the one about the machine needed to complete the correspondent task (color W). The place node P2 stores the tokens with the logical sequence information about the jobs in place P1 (colors E and W) and the time consumption when the task is done (color P). The last place node P3 represents the availability of the different machines (color Z). In the initial state of the system all jobs are waiting for the first task to be done, and the machines are available and ready to be used. The global clock and the time stamps are equal to zero in this state.

3.2. Second Step: Performance evaluation

The aim of the data collected is to evaluate the efficiency of the optimization process while the time stamp analysis is carried out. During the simulation, data of the following events was collected:

- The amount of repeated states (old nodes) that appeared during the exploration.
- The number of old nodes that were updated due to the comparison of their time stamp combination.
- Number of not-updated nodes (rejected nodes) due to the comparison of time stamp values.
- Number of old nodes for which it was not possible to decide about updating based on time stamp comparison. In order to come to a decision, a depth exploration was done for these cases.
- The number of nodes that needed to be updated after the depth exploration was done (because the final time was less than the original one).

The results obtained from the evaluation of the different benchmarks are presented in Table 1.

Table 1: The Simulator Performance

Job-Shop type	J-S 3x3	J-S 5x5	J-S 6x6
No. of different States	693	7,776	117,650
Old Nodes Found	2,032	24,625	487,408
Updated Nodes at Initial Evaluation	59	1,893	26,555
Discarded nodes after Initial Evaluation	485	5,829	128,387
Nodes Unable to Decide	1,488	16,903	332,475
Updated Nodes after depth explorations	11	10	35

The above data was collected during the optimization of the different benchmarks, maintaining in memory all the information for the complete state space and choosing the first available node as a depth-first search logic.

3.3. Analysis of the Experimental Results

It can be seen that in the case of the job-shops, the percentage of updated nodes after the depth exploration is 0.7%, 0.06% and 0.01% for the 3x3, 5x5 and 6x6 job-shop respectively. The previous results are important since the optimization phase takes a lot of CPU time, which can be reduced by minimizing the number of depth explorations.

It also can be seen that the time analysis becomes more complicated and the number of states which are unfeasible to decide, increase dramatically as the model becomes more complex (from around 1,500 to 16,000 and 300,000 for each of the benchmarks). Based on the result interpretation, it can be concluded that the simulator performance could be improved if the amount of irresolute nodes was reduced. The reduction was achieved through an heuristic developed for making a better selection of child nodes during the exploration of the state space.

4. IMPROVING THE EXPLORATIONS

The results presented in Table 1 show that the time analysis performance is not as efficient as it could be (it only updates very few states after making the depth exploration). A better way of selecting the nodes for the exploration is proposed. The selection rule aims to choose the best candidate among child states when the next depth level is to be explored. Selecting a better node will increase the number of rejected nodes, and the number of depth explorations will be reduced when the evaluation of old nodes takes place. In addition, it is considered that choosing good nodes during the exploration will lead to the best path in a faster way and therefore the optimization time spent in the second phase of the algorithm is reduced.

4.1. An Heuristic for Selecting the Best Nodes

The heuristic implemented was coded with the purpose of observing the improvement that can be reached if a better selection of states to be evaluated is done. A utility function that allows assigning a value for each one of the states was constructed. The function was developed on the basis that the less value the time stamps have, the less probable incrementing the clock is for the subsequent evaluations of any marking.

That is, given a M_j state of the reachability tree, let F be the function that sums the time stamps Ts_i where the index i goes from token 1 to token N_j of the marking.

$$F(M_j) = \sum_{i=1}^{N_j} Ts_i \qquad (1)$$

During the exploration of the state space, the selection of a node among child nodes is done based on the value calculated by the function (1) and the node which has the lowest value is the one to be selected. In other words, given a set of k child nodes, the next node among them to be evaluated is selected under the next criteria:

$$Min\left\{F(M_j) \ni j = 1,\ldots,k\right\} \qquad (2)$$

The index j represents the child nodes of a given state.

An example of how to use the utility function to select the best child is presented in Figure 5.

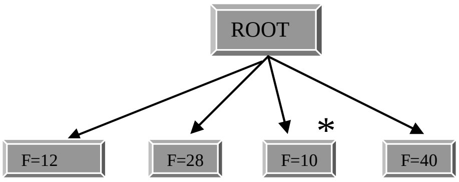


Figure 5: Using the Utility function to select the Best Node

In this figure, there is a root node with four different child nodes. If there were no utility function implemented, the first child to be evaluated would be the node on the left hand side (F=12). Using the utility function, a value assignment to each of the different child nodes is made; the child to be evaluated is the one with the lowest utility value. In this example the node with the value of 10 units is the chosen one. Therefore the utility function is used to guide the depth-first exploration algorithm for the generation of the reachability tree.

The heuristic was implemented in the algorithm, the models were run again and the data was collected to evaluate the new performance. The results obtained using this algorithm are presented in Table 2.

Table 2: Simulating with the heuristic

Job-Shop type	J-S 3x3	J-S 5x5	J-S 6x6
No. of different States	693	7,776	117,650
Old Nodes Found	2,032	24,625	487,408
Updated Nodes at Initial Evaluation	0	1,330	18,725
Discarded nodes after Initial Evaluation	2,032	10,506	213,966
Nodes Unable to Decide	0	12,789	254,717
Updated Nodes after depth explorations	0	22	34

4.2. Results

From the results presented in table 1 and 2, it can be seen that using the utility function, the selection rule is improved and as a result the performance of the exploration task is drastically enhanced. It can be seen that for the 3x3 job-shop the selection of the best path is achieved at the first step of the algorithm. This conclusion takes place since during the second phase of the algorithm all stored nodes are discarded in the old node list. Therefore it is not necessary to make a depth exploration to decide whether a time stamp updating is needed or not. In this particular case, the chosen states for opening the reachability tree are the best ones. From the results of the rest of the models (5x5 and 6x6 job-shop), it can be seen that the number of updatings practically remains the same. The analysis of the number of discarded nodes, allows concluding that the utility function improves the simulation performance.

As a result of using the heuristic, the discarded nodes increase by an amount of 5,000 for the 5x5 job-shop and by around 100,000 for the case of the 6x6 job-shop. The avoidance of the same quantity of depth explorations during the analysis of old nodes gives as a result a better performance of the simulator. The same conclusion is drawn by the analysis of the irresolute nodes. In the case of the 5x5 and 6x6 job-shops the numbers of depth exploration is drastically reduced and the performance of the simulator is improved since less depth explorations are necessary.

4.3. Conclusions and Future Work

The reachability tree-based simulator presented in this article is a promising approach for solving real life industrial problems. It has been introduced an heuristic implemented for the first step of the algorithm which leads to exploring the reachability tree in a better way by reducing the quantity of evaluations needed in the second step. In this way time optimization is achieved. It has been also demonstrated that focusing on the best selection of nodes during the exploration of the reachability tree improves drastically the simulator performance. Due to the results presented, the development of an heuristic focusing on the selection of old nodes to be evaluated during the second step of the algorithm is proposed as a future work. It is expected that implementing such an heuristic in combination with the one introduced in this article will further increase the performance of the whole algorithm.

REFERENCES

Christensen, S., Kristensen, L.M., Mailund, T., 2001“A Sweep-Line Method for State Space Exploration“in *TACAS*, Springer-Verglag, Berlin-Heidelberg

Christensen, S, Mailund T., 2002, “A Generalized Sweep-Line Method for Safety Properties”, in FME, Springer- Verlag, Berlin- Heidelberg

Dauzére-Peres, S., Lasserre, J.B., 1994, “An Integrated Approach in Production Planning and Scheduling”, *in Lecture Notes and Mathematical Systems*, Springer-Verlag, Berlin.

Holzmann, G.J., 1998, “An analysis of Bitstate Hashing”, in *Formal Methods in System Design*, V13 (3), p.p.301-314, November.

Jard C., Jeron, T., 1991”Bounded –memory algorithms for Verification On-the Fly”, in *Proceedings of CAV 1991*, vol575 of *Lecture Notes in Computer Science*.Springer-Verlag.

Jensen, K 1997. “Coloured Petri Nets: Basic Concepts, Analysis Methods and Practical Use”. Vol. 1 Springer-Verlag. Berlin.

Jensen, K, 1997, “Coloured Petri Nets: Basic Concepts, Analysis Methods and Practical Use”, Vol 2, Springer-Verlag. Berlin

Mujica, M., Piera, M.A., 2006 ”Building an Efficient Coloured Petri Net Simulator”, in *Proceedings of the International Mediterranean Modeling Multiconference,* p.p.153-158, October 4-6, Barcelona, Spain

Mujica, M., Piera, M.A, 2007, ”Data Management to Improve CPN Simulation Performance”, in *Proceedings of the International Mediterranean Modeling Multiconference*,p.p.53–58, 4-6 October, Bergeggi, Italy

Mujica, M., Piera, M.A., 2007,”Improvements for a Coloured Petri Net Simulator”, in *Proceedings of the 6th EUROSIM Congress on Modeling and Simulation*, p.p. 237, September 9-13, Ljubljana, Slovenia

Narciso, M., Piera, M.A., y Figueras J., 2005, “Optimización de Sistemas Logísticos Mediante Simulación: Una Metodología Basada en Redes de Petri Coloreadas”, Revista Iberoamericana de Automática e Informática Industrial, vol 2(4), p.p. 54-65IAII, Valencia, España.

FIRST PRINCIPLES MODELING OF THE LARGE HYDRON COLLIDER'S (LHC) SUPER FLUID HELIUM CRYOGENIC CIRCUIT

Rafal Noga[a], **Cesar de Prada**[b]

[a] Cryogenics Group at AT Department, CERN; Room 36-2-002; 23 Geneva 1211; Switzerland
[b] Department of Systems Engineering and Automation; Universidad de Valladolid; Facultad de Ciencias; c/ Real de Burgos s/n. 47011Valladolid; Spain

[a] rafalnoga@)hotmail.com, [b] prada@autom.uva.es

ABSTRACT

The Large Hydron Collider (LHC) at CERN is the largest particle accelerator in the world. It uses more than 1600 superconducting magnets that will be maintained at operational temperature equal to 1.9K by Super fluid Helium Cryogenic Circuit. This document presents a first principle nonlinear model of the circuit to be used in the development process of a Nonlinear Model Predictive Control (NMPC) of LHC Standard Cell temperature. First, general information on the context of the project and the motivation for the development of the model are presented. Then, the Super fluid Helium Cryogenic Circuit is described with details and the modeling process of the circuit is presented step by step. Finally, the model is adjusted with the experimental data, the results of the model simulation are evaluated and the future work is proposed.

Keywords: LHC, Standard Cell, super fluid helium, cryogenics, first principle modelling

1. INTRODUCTION

The basic functions of beam guiding and focusing in the Large Hydron Collider (LHC) at CERN are performed by more than 1600 superconducting magnets in order to produce strong magnetic fields..

The NbTi windings of the magnets must be maintained at operational temperature equal to 1.9K in order to minimize energy losses maintaining the cables under superconducting conditions. This is done by the Super fluid Helium Cryogenic Circuit

The experiments, performed on the String and String2 LHC Prototypes from 1995 to 2003, showed that the regulation of the LHC magnets temperature is a challenge, mainly due to presence of strong nonlinearities together with frequent changes of the operation point, inverse response on the control input manipulation and variable dead time of the response.

PID controllers used to control the temperature have poor performance. A MPC controller was developed, tested and showed a great potential to improve the temperature control of the first 53 meter long "Half Cell" String Prototype. Unfortunately the MPC controller could not be commissioned on the 107 meter long "Full Cell" String2 Prototype because of the mismatch between the distributed parameter process dynamics and its lumped parameter model used in the controller.

This document presents development of a new, “first principle" nonlinear model of the Super fluid Helium Cryogenic Circuit of a LHC Standard Cell to be used in the process of development of a Nonlinear Model Predictive Control (NMPC) of the superconducting magnets temperature.

In the next section, the Super fluid Helium Cryogenic Circuit is described and some aspects of the modeling are presented. Then, the model is adjusted with the experimental data from the String2 experiment. Finally, the results of the model simulation are evaluated and the future work is proposed.

2. SUPERFLUID HELIUM CRYOGENIC CIRCUIT OF A STANDARD CELL

The task of the Super fluid Helium Cryogenic Circuit of a LHC Standard Cell is to stabilize the windings temperature of eight magnets comprised in the Standard Cell. This is done by placing the magnets into a static bath of pressurized super fluid helium that has very high thermal conductivity and penetration abilities. The heat is removed from the bath by a 107m long bayonet heat exchanger (BHX) running along the bath, with saturated helium flowing inside and evaporating thus providing cooling, see figure 1.

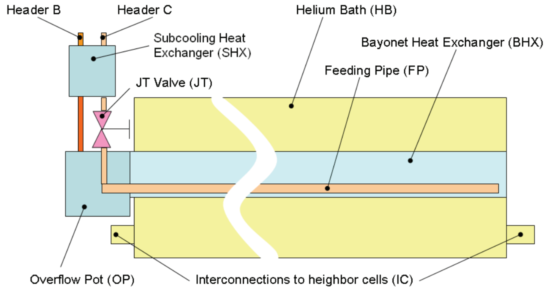

Header B
Header C
Subcooling Heat Exchanger (SHX)
Helium Bath (HB)
Bayonet Heat Exchanger (BHX)
Feeding Pipe (FP)
JT Valve (JT)
Overflow Pot (OP)
Interconnections to heighbor cells (IC)

Figure 1: Simplified scheme of the Standard Cell Cryogenic Circuit

The liquid helium to be used in the BHX is taken from the header C at 4.6 K and 3 bar, cooled down to

2.2K in a sub-cooling heat exchanger (SHX), in counter-flow with 1.8 K, 16mbar helium vapor leaving the BHX. Then the pressurized He II at 2.2K, 3bar is expanded to about 16mbar via the assembly of a Joule-Thompson (J-T) valve and a small diameter feeding pipe (FP) placed inside the BHX, that deposits both the saturated helium and the vapor created during the expansion at the opposite (to the JT valve) end of the BHX. Then the saturated He II flow inside the BHX in opposite direction, towards the JT valve evaporating and absorbing heat from the HB.

The flow of evaporating helium, thus the amount of heat that it can absorb, is regulated via the JT valve position. The BHX is sized to work as partially wetted - helium flowing in the exchanger evaporates completely before reaching the opposite end of the BHX.

The vapour at about 1.8 K created in the BHX together with the vapour created during the expansion in the JT-FP assembly is pumped out to the Header B in order to keep the very low vapour pressure (16 mbar) in the BHX, that determines the saturation temperature of super liquid helium flowing through the BHX, thus the minimal achievable temperature of the helium bath.

The description is illustrated with a simplified scheme, see Figure 2.

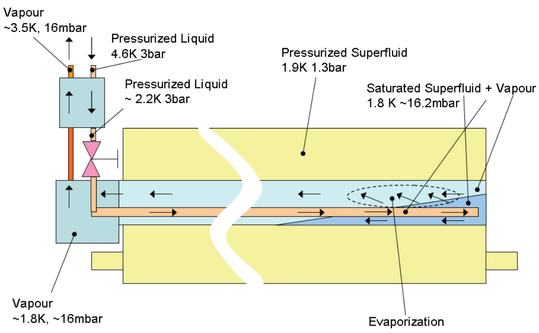


Figure 2: Simplified helium flow in the Standard Cell Cryogenic Circuit

The temperature of the HB in the String2 Phase 3 Prototype was measured at 8 points along the Standard Cell corresponding to each magnet position.

Experiments performed during the String Experimental Programme showed that the circuit dynamics includes: asymmetric inverse response of the bath temperature on a JT valve manipulation (where temperature excursion varies in function of its direction), variable dead-time depending mostly on the heat load situation, and non-uniform magnet temperature across the string of magnets due to a constrained longitudinal heat transfer (Chorowski 1998).

3. MODELING THE SUPERFLUID HELIUM CRYOGENIC CIRCUIT

The purpose of the model is to enable prediction of future temperature values of the Standard Cell Helium Bath that is defined by heat flows in the magnets, their geometry and material properties. After few significant simplifications of the magnet representation, like considering a uniformly distributed corrected mass of helium, its entropy can be calculated as:

$$\frac{\partial(\rho H(x,t))}{\partial t} = \frac{\partial q_l(x,t)}{\partial x} + \frac{\partial q_t(x,t)}{A} \qquad (1)$$

where q represent longitudinal and transversal heat flow throw the HB.

Using a finite volume approach for spatial discretization of the one-dimensional model of the magnets the change of enthalpy of a discrete segment of the bath is proportional to the sum of all heat fluxes interchanged with the sector:

$$\frac{d(H_i(t)m_i)}{dt} = \sum Q_i(t); i \in < 1...20 > \qquad (2)$$

Relation between the enthalpy and temperature has been tabularized.

The heat transfers, illustrated in the Figure 3, are:

- static heat loads into the HB that originate at ambient temperature
- dynamic heat loads into the HB originating at magnet components
- heat exchanged between the HB and BHX that cools the HB
- heat flowing from the FP into the HB, responsible for the asymmetric inverse dynamic response of the cooling loop
- longitudinal heat flow through the HB due to its non uniform temperature at the interface of two HB sectors
- heat exchanged through interconnections at the extremities of the HB;

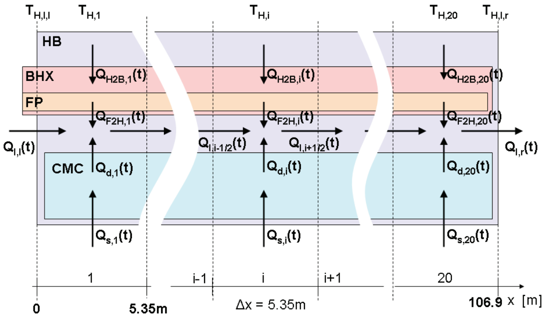


Figure 2: Heat flow in the Standard Cell Cryogenic Circuit

3.1. Counter-flow heat exchange in super fluid helium

Longitudinal Heat Transfer in the bath of HeII is very effective due to the enormous heat conductivity of super fluid helium at moderate heat flux. The mechanism of heat transfer in super fluid has been described (Bottura and Rosso 1999). In the one-dimensional case like for the HB, using the pseudo conduction term, called Super fluid Thermal Conductivity Function F(T;p), the heat flux q in the super fluid can be related to the temperature gradient

$$q'' = -F(T,p)\frac{\partial T}{\partial x} \tag{3}$$

Between the heat flux and the temperature gradient exist a stiff bidirectional relation as temperature of the HB sector if a function of heat flows. This causes instabilities in the numerical integration of the simulation as the gradient goes to 0. For this reason a modified expression for the longitude heat flow between discrete sectors is used in the model:

$$Q_{l,i} = \begin{array}{l} -A(F(T)\frac{dT_i}{dx})^{0.33}; \frac{dT_i}{dx} > 1e-7 \\ -Aconst.\frac{dT_i}{dx}(F(T))^{0.33}; \frac{dT_i}{dx} < 1e-7 \end{array} \tag{4}$$

3.2. Heat transfer to the Bayonet Heat Exchanger. Kapitza conductance

The heat is evacuated from the HB to the Bayonet Heat Exchanger by conduction across the wall of the 107m long Bayonet Heat Exchanger (BHX) tube running all along the Standard Cell that is limited by the transversal thermal conductance C_{th} between HB and BHX:

$$Q = (T_1 - T_2)C_{th} \tag{5}$$

that is dominated by Kapitza conductance of liquid-solid interfaces justifying neglecting of the bulk thermal resistance of BHX wall

$$C_{th} = C_K A T^3 \tag{6}$$

where the Kapitza coefficient C_K has been measured for the BHX.

Finally the heat transfer between the HB and the BHX for the i-th sector of the Standard Cell depends on the surface A corresponding to wetted area of the BHX.

The investigation of the II-phase helium flow in the BHX showed that the saturated HeII would mostly operate in the stratified flow regime (Lebrun 1997) thus the wetted area of a BHX sector can by directly calculated from the level h (corresponding to mass) of helium present in the BHX sector:

$$A = \Delta x 2r \arccos(\frac{r-h}{r}) \tag{7}$$

4. MODEL ADJUSTING AND SIMULATION

Model contains nine coefficients representing unknown model parameters determining its dynamics. Values of seven coefficients have been estimated by minimalization of a quadratic error function, calculated based on differences between simulated values of magnet temperatures and the corresponding experimental data from the String2 Phase3 experiment. Two other parameters have been hand fitted, because no specified, accurate error functions have been defined as far.

4.1. Restoring the experiment configuration

During the experiment the extremities of the Standard Cell instead to other cells were connected to an electrical feed-box DFBX and to a magnet return box MRB, both introducing additional heat flows throw the interconnections that have to be modeled.

The static heat load present during the String2 experiment has to be estimated in order to set the operational point of the simulation. It is done in a very simple way using an observer containing the model of the cryogenic circuit. In the observer configuration, the control input of the model is the static heat load, to be estimated, and the outputs are the magnet temperatures. The regulation error of the temperature is feed into the PI controller that calculates the estimated value of the static heat load.

4.2. Experimental data accuracy

The data coming from three String2 experiments has been used to validate the model. The performed experiments consisted on the JT valve manipulation (experiment “Nominal”), introducing additional heat to the magnets (experiment “Heatload”), and both actions performed sequentially (experiment “Mixed”).

The accuracy of the special thermometers measuring the magnet temperatures below 2K is +/- 10mK, which is comparable to small variation of magnets temperatures. To improve the accuracy of the simulation the offset of the temperature measurement is estimated by observation of the temperatures in case when the BHX is full of saturated helium and all the temperatures should be equal.

4.3. Sequential estimation of model parameters

The model parameters have been optimized sequentially one after another using NAG single variable optimization routine e04abc (based on quadratic interpolation), (Nag 2008). The optimal coefficient value in each iteration has been found with relative tolerance

$$Tol(\bar{x}) = 1\% \tag{8}$$

The sequential optimization allowed precise optimization with quadratic error functions specified separately for each parameter, taking into account the temperature error only at points where the parameters influence is most visible. For example, the total mass of helium in the magnets has been estimated regarding only the increase rates of magnets temperatures in response on additional heat load introduced during the experiment.

The order of optimization of the parameters is very important, because some parameters influence significantly the temperature response of the model in all operating states, and other only at some specified points. The parameters with wider influence on the model behavior, that stronger influence the optimal values of other parameters, are optimized first. For example, the total mass of helium in the magnets that has been mentioned above has been estimated at the beginning, because it determines the general dynamics of the model.

That is the reason, why the experiment with an extra heat load is used first during the optimization – the absence of valve manipulation during the experiment makes possible optimization of many model parameters, at the same time excluding errors related to inappropriate modeling of super fluid helium dynamics, will be optimized later in the sequence.

The model parameters have been estimated in following order (see the Figures 4 and 5, where the described below zones of interest of model temperature response are pointed),

1. effective cross-section of the magnet determining the values of longitude counter-flow heat flux, regarding static temperature distribution
2. effective friction factor of the very low pressure channel of the sub-cooling heat exchanger, influencing the pressure in the BHX, thus the temperature of the saturated helium in the heat exchanger
3. effective mass of super fluid helium in the magnet, described in an example above
4. rangeability value of the JT valve, determining the estimated value of the static heat load, has been estimated regarding the total energy rest in the magnets after the experiment, thus the the differences between magnet temperatures at the beginning and at the end of experiment
5. effective viscosity parameter of the super fluid helium in the Bayonet Heat Exchanger has been found based on observation that in the experiment with additional heat load, the dynamics of magnet temperature response at the end of the magnet string opposite to the JT Valve, where the super fluid helium is deposited, depends on the static distribution of saturated helium in the bayonet heat exchanger tube, thus on its effective viscosity
6. effective heat conductivity between Inner Pipe and the Helium Bath, determining the heat flow between them, and
7. effective friction of the two-phase helium flow in the Inner Pipe, determining the pressure distribution along the pipe, have been found regarding the inverse response of the magnet temperature during JT Valve manipulation.

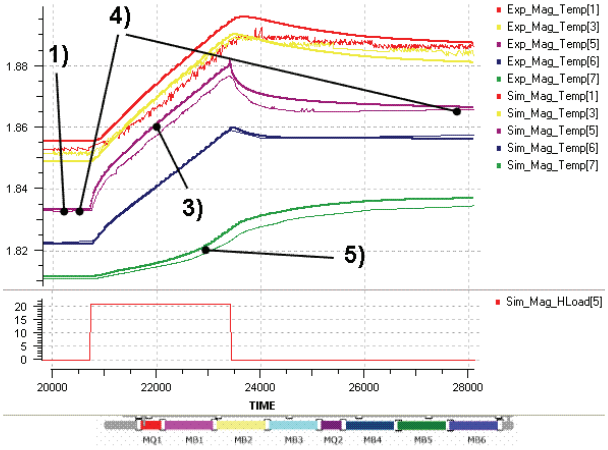


Figure 4: Model simulation: experiment « Heat Load»

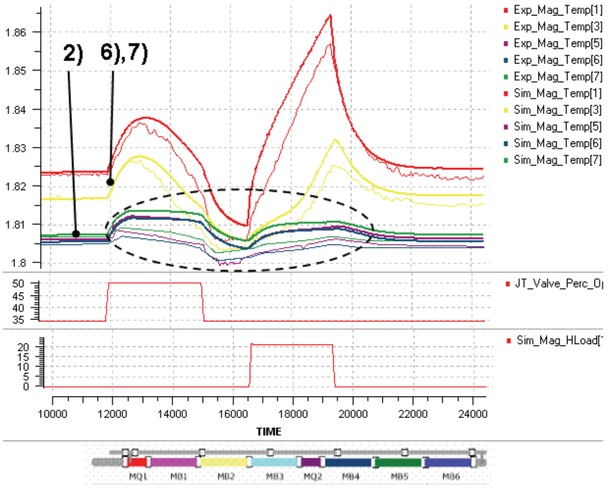


Figure 5: Model simulation: experiment « Mixed»

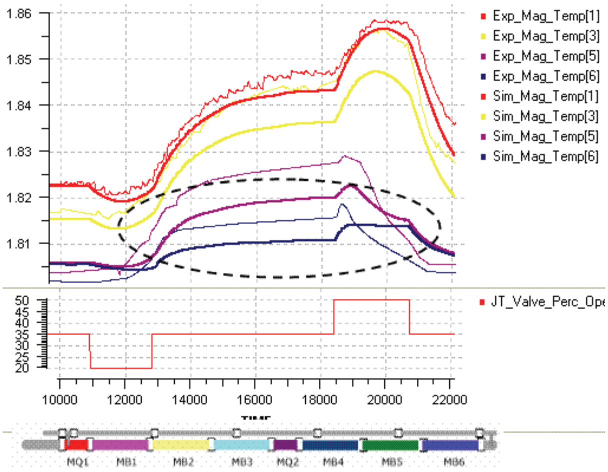


Figure 6: Model simulation: experiment « Nominal»

5. RESULTS AND FUTURE WORK

Model has been validated with three sets of experimental data corresponding to the three experiments performed on the String2 Prototype. This data is reliable for model validation because it represents distinct operational scenarios, moreover the data from experiments “Mixed” and “Nominal” has not been used to model adjusting.

The results of simulation (bold lines) and the experimental data (thin lines) for the three experiments are presented in the Figures 4, 5 and 6.

The simulation results are very close to the experimental data, but some problems remain unsolved.

Probably the wrong modeling of the pressure profile inside the pipe feeding helium inside the BHX causes inappropriate changes in vapor mass flow inside the BHX that could be the reason for temperature simulation errors in magnets opposite to the JT Valve, see areas encircled by dashed lines in Figures 5 and 6.

The way of calculation of helium temperature after sub cooling in the SHX has not been already found, thus experimental values are used during the simulation and model cannot be used to predict magnet temperatures.

ACKNOWLEDGMENTS

The model has been developed in frame of the Master Thesis Final Project under name "Modeling and control of the String2 LHC Prototype at CERN" and PhD Project under name “Non-linear Model based Predictive Control (NMPC) of the Large Hydron Collider’s (LHC) Super fluid Helium Cryogenic Circuit” in 2007 and 2008. The authors wish to express their gratutude for the support provided by project DPI2006-13593 and CERN

REFERENCES

Blanco, E., 2001. *"Nonlinear Model-based Predictive Control Applied To Large Scale Cryogenics Facilities"*. PhD thesis. Universidad de Valladolid

Bottura, L., Rosso, C., 1999. "Finite element simulation of steady state and transient forced convection in super fluid helium". *International Journal for Numerical Methods in Fluids* 30: 1091–1108.

Camacho, D., 1998. *"Thermal Characterization of the HeII LHC Heat Exchanger Tube"*. CERN. Available from: http://doc.cern.ch/archive/electronic/cern/preprints/lhc/lhc-project-report-232.pdf [July 20, 2008].

Chorowski, M., 1998. "A proposal for simplification of the LHC cryogenic schema", *Advanced Cryogenics Eng., A* 43: 395-402.

Flemster, B., 2000. *"Investigation, Modelling and Control of the 1.9 K Cooling Loop for Superconducting Magnets for the Large Hadron Collider"*. PhD thesis. NTNU Trondheim, Institutt for klima- og kuldeteknikk IX

Flynn, T., 1997. *"Cryogenic Engineering"*. New York: Marcel Dekker Inc.

Gorter, C.J., Mellink, J.H., 1949. "On the irreversible processes in liquid helium II", *Physica* 15: 285304

Lebrun Ph., 1997. *"Cooling Strings of Superconducting Devices below 2 K: The Helium II Bayonet Heat Exchanger"*. CERN. Available from: http://doc.cern.ch/archive/electronic/cern/preprints/lhc/lhc-project-report-144.pdf [July 20, 2008].

NAG, 2008. *“nag opt one var no deriv (e04abc)”*. NAG. Available from: http://ftp.nag.co.uk/numeric/CL/manual/pdf/E04/e04abc.pdf [July 20, 2008].

Ödlund, E., 2007. *"Virtual instrumentation: Introduction of virtual fow meters in the LHC cryogenics control system"*. MSc thesis. Linkopings Universitet

Prada, C., Blanco, E., Cristea, S., Casas, J., 2004, “Nonlinear Predictive Control in the LHC”. *7th IFAC Int. Symp. on Advanced Control of Chemical Processes, ADCHEM'03*, 442-448. , Jan. 2004, Hong-Kong, China

AUTHORS BIOGRAPHY

Rafal Noga; Ph.D. student at the University of Valladolid, Spain. Actually works on the project “Non-linear Model based Predictive Control (NMPC) of the Large Hydron Collider’s (LHC) Super fluid Helium Cryogenic Circuit” at CERN, Switzerland.

Cesar de Prada; Professor of Systems Engineering and Automatic Control with the University of Valladolid, Spain. Current topics of interest include modelling and simulation, predictive control and process optimization.

SOLVING THE PALLET LOADING PROBLEM USING A COLOURED PETRI NET APPROACH

Miquel Angel Piera$^{(a)}$, Catya Zuñiga$^{(b)}$

$^{(a)\,(b)}$Autonomous University of Barcelona, Dept. of Telecommunication and Systems Engineering, Barcelona, Spain

$^{(a)}$miquelangel.piera@uab.es, $^{(b)}$ CatyaAtziry.Zuniga@uab.cat

ABSTRACT

There is a lack of commercial decision support tools that could help to deal with the best configuration of decision variables to optimize the performance of a system with a stochastic, dynamic and synchronous behaviour. In this paper a Coloured Petri Net model that formalizes the Pallet Loading problem and its optimization by integrating evaluation methods with search methods will be presented.

Most Pallet Loading optimization tools, provide very good solutions when the number and type of boxes to be placed in a pallet is considerably low, which is the case of most production industrial systems, since they are used to work with a reduced number of master boxes. However, there are other systems (ie. distribution warehouses) in which there are a high diversity of boxes to be placed in pallets. The model developed provides very good results by using certain heuristics that avoid the analysis of the whole search space, evaluating only the best scenarios.

Keywords: Coloured Petri Nets, Constraints, Coverability Tree, Simulation

1. INTRODUCTION

World-wide market competition, short cycle product time, together with random demands instead of steady demands, are some key-factors which have forced industry to change traditional rigid and/or non-automated production architectures (such as Flow Shop, Job Shop) towards Flexible Manufacturing Systems (FMS).

Despite flexibility to react to market fluctuations can easily be achieved by reprogramming production units (CNC machines), and transport resources, efficient flexibility can only be achieved by a correct coordination of all the entities (material and resources), that takes part on the production and transport processes. A key factor in this highly competitive market is the ability to respond rapidly to changes in the demand while minimizing costs.

Most research and development efforts have been focussed in the process of planning the efficient, cost effective flow and storage of raw materials, in-process inventory, finished goods and related information from point of origin to point of consumption. However, the picking of the final deliverable products into pallets can be a critical supply chain component for many distribution centers.

The complexity in transport, the changing mentality in logistics and the constant need to improve competitiveness provoke the necessity of developing new tools that could tackle the problem from a global point of view, considering operational, strategic and tactic decisions.

There are different methodologies that have been used traditionally to give response to planning problems. Modelling and Simulation techniques have proved very useful in strategic and tactic design. However, several limitations appear when trying to find a feasible solution to a NP-complete problem. A limited number of scenarios can be evaluated in an acceptable time interval.

A solution for these operational decision problems would be finding a modelling approach that could tackle these problems by integrating different approaches: search methods (AI, OR) with evaluation methods (Simulation).

Coloured Petri nets provide a framework for the construction and analysis of distributed and concurrent systems. A CPN model of a system describes the states which the system may be in and the transitions between these states. CP-nets have been applied in a wide range of application areas, and many projects have been carried out in industry. In contrast to most specification languages, Petri nets are state and action oriented at the same time – providing an explicit description of both the states and the actions, (Kristensen, 1998).

A Discrete Event System model has been developed to formalize the pallet packing problem (PLP), using the CPN formalism to analyze the best configuration that minimizes the number of pallets required for packing a certain amount of boxes.

2. A DISTRETE EVENT MODEL FOR PALLET MAKER

There are several modelling approaches to tackle the PLP, some of them are based in container loading approach which try to construct vertical 'walls' across the container (Lim, Rodrigues, and Yang, 2005). George and Robinson (1980) where the first to formulate a heuristic algorithm called the wall-building method for the packing of up to 20 different box types into one container.

Gehring, Menschner, and Meyer (1990), proposed a method to pack rectangular boxes of different size into a shipping container of known dimensions. This problem specification does not allow for orientation constraints. The algorithm is also based on the wall-building method. In this method, a series of ranking rules such as box ranking rule, box position ranking rule are assumed. Bischoff and Dowsland (1982), developed a method based on filling the container by building layers across its width. Han, Knotta and Egbelu (1989) provided a heuristic based on the dynamic programming method and wall-building method. In his study, the objective was to maximize the volume occupancy without no orientation constraints. The algorithm calls for the boxes to be packed along the base and one vertical wall of the container. After a L-shaped packing is complete, a "new" container can be formed, which become the focus of the successive packing. Xue and Lai (1997), provided another algorithm based on the wall-building method, in which both the cargo and container must be rectangular. This algorithm integrated three heuristics:

- The ordering heuristic sorts the cartons according to the depth, quantity and surface area. Higher priority was assigned to cartons with larger values of the above characteristics.
- The placement heuristic determined the depth of the new layer.
- Layer-building heuristic.

Ngoi, Tay, and Chua (1994), designed a heuristic algorithm based on "spatial representation" techniques to solve the problem of packing rectangular carton boxes into a single rectangular container, which includes finding the best placement position and orientation for each box. By increasing the number of boxes to be fitted in the container, the matrix expands to accommodate additional information. This packing algorithm is independent of the ordering of the boxes which usually exists in other algorithms. However, the strategy divides the original container into many small empty spaces which are not suitable for holding big boxes in successive packing. Chua, Narayanan, and Loh (1998) described an improved method based on Ngoi, Tay, and Chua (1994), which allows the user to suggest the locations of certain boxes which exists in real-life packing problems.

Ivancic, Mathur, and Mohanty (1989), proposed an integer programming based heuristic approach to the three dimensional packing problem in which a container is packed considering to maximize a linear function of the boxes packed.

Most of these packing strategies were based on the wall building approach with some adaptations. The method itself is a greedy heuristics which can lead to weak final solutions. Other methods include using spatial representation, reducing the problem to maximum clique problem or an integer programming problem. For methods based on the spatial representation, empty volume search routines are used to generate small spaces, whereas the successful utilization of these small spaces is difficult. For methods based on graph theory and integer programming, some disadvantages appear due to model simplification aspects and the inherent difficulty of the combinatorial nature of these problems. On the whole, published works generally give successful implementations and provided some interesting insights into the various views on how successful packings can be best achieved, (Lim, Rodrigues, and Yang, 2005).

The PLP considered in this paper deals with n different types of boxes (rectangular shaped) with known dimensions cxi, cyi, czi (with i = 1 . . . . . n) which have to be placed onto a surface pallet of size scx * scy. The objective is assumed to be the maximisation of the number of boxes to be fitted inside the pallet, preserving (i.e. the maximisation of pallet utilisation) preserving several constraints such as the maximum weight supported in each pallet, the maximum height and floors recommended, etc.

Boxes may be rotated 90º so long as they are placed with edges parallel to the pallet's edges, i.e., the placement must be orthogonal, and they can be rotated only once. It is assumed, without loss of generality, that cxi, cyi, czi, scx, scy, gz, are positive integers, in which scx, scy defines the pallet position where the left hand corner of the box-i has been placed, and gz corresponds the pallet floor where box-i has been placed. It is also assumed that each box has a ''this side up'' restriction.

The palletizing problem can be seen as a DES by busing a proper abstraction level in which each event represents the placement of a certain box in the pallet surface, as it is represented in Figure 1.

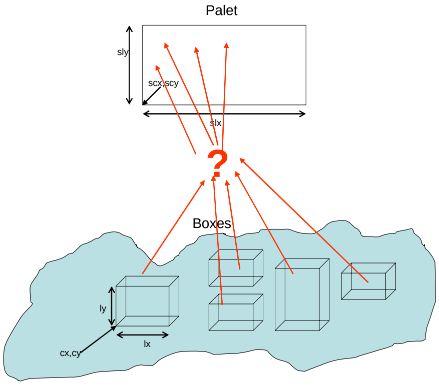


Figure 1: Placing boxes into a pallet

Attributes can be defined to describe the pallet configuration: the coordinates of each box placed in the pallet surface together with the coordinates of the fragmented space as the results of placing a box in the pallet.

As a consequence of placing a box in the pallet, fragmented surfaces will appear. Figure 2 illustrate this situation, in which it is easy to identify two different surface areas in the pallet that should be evaluated to fit the next box in the pallet. Thus, events describing different possibilities for placing a box in a pallet should compute the new layout configuration of the pallet once the box has been placed: position and orientation of each box in the pallet, together with the computation of the dimensions of each new free fragmented space generated as a consequence of placing boxes.

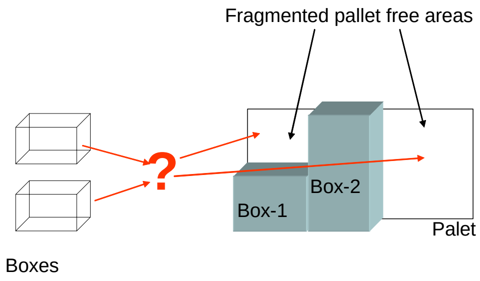


Figure 2: Fragmented areas in a pallet

3. MODEL

The Pallet loading model has been specified in the CPN formalism (Jensen, 1997) and it can be decomposed into 5 different set of transitions:

- Transitions T_{21} … T_{24} : These transitions represent fitting a box into a free pallet area where the length *x* and/or *y* of the box are equal or shorter than the length *x* and *y* of the free pallet surface
- Transitions T_{31} … T_{35} : These transitions represent fitting a box into a free pallet area where the length *x* and/or *y* of the box are longer than the length *x* and *y* of the free pallet surface
- Transitions T_{41} … T_{48} : These transitions represent fitting a virtual box into a free pallet area where the length *x* and/or *y* of the box are equal or shorter than the length *x* and *y* of the free pallet surface
- Transitions T_{51} … T_{58} : These transitions represent fitting a virtual box into a free pallet area where the length *x* and/or *y* of the box are longer than the length *x* and *y* of the free pallet surface

3.1. Colour specification

Table 1 and 2 summarizes the colours and places used to describe all the information required to fit boxes in a pallet using the abstraction level of the pallet maker process introduced in this section.

Table 1: Place specification

Place	Colour	Meaning
P1	1'(idc,cx,cy,cz,lx,ly,lz,cr,ce)	Boxes
P2	1'(scx,scy,slx,sly)	Free Pallet Surfaces
P3	1'(idc,cx,cy,cz,lx,ly,lz,ce)	Virtual Boxes
P4	1'(ge,gz,gsf,gncv,gnc)	Global Variables

Table 2: Colour specification

Colour	Definition	Meaning
idc	Integer	Box identifier
Cr	Integer	0: original orientation 1: rotated 90º wrt Z
Ce	Integer	0: not assigned 1: working 2: placed in the pallet
Cx	Real	Coordinate X where the box is located
Cy	Real	Coordinate Y where the box is located
Cz	Real	Coordinate Z where the box is located
Lx	Real	Box length in coordinate X
Ly	Real	Box length in coordinate y
Lz	Real	Box length in coordinate z
Scx	Real	Coordinate X where the surface is located
Scy	Real	Coordinate Y where the surface is located
Slx	Real	Surface length in coordinate X.
Sly	Real	Surface length in coordinate Y.
ge	Integer	0: A Box can be placed in the pallet 1: Box to be assigned 2: Looking for a surface 3: Evaluating the new fractioned surfaces
gz	Integer	Indicates the pallet floor
gsf	Real	Available surface in the pallet
Gncv	Integer	Number of virtual boxes
Gnc	Integer	Total number of boxes

3.2. Events examples

Figure 4 illustrates the event that formalizes fitting a box into a free pallet area if the length *x* and *y* of the box are shorter than the length *x* and *y* of the free pallet surface. It should be noted that under these circumstances, the fragmented new spaces in the pallet is incremented in two new free areas that can be used as free pallet areas to place future boxes.

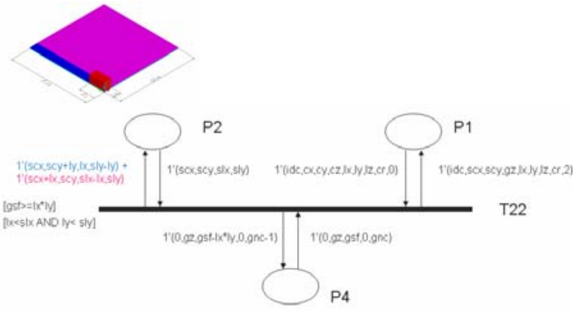

P2
P1
1'(scx,scy+ly,lx,sly-ly) +
1'(scx+lx,scy,slx-lx,sly)
1'(scx,scy,slx,sly)
1'(idc,cx,cy,cz,lx,ly,lz,cr,0)
1'(idc,scx,scy,gz,lx,ly,lz,cr,2)
[gsf>=lx*ly]
[lx<slx AND ly< sly]
T22
1'(0,gz,gsf-lx*ly,0,gnc-1)
1'(0,gz,gsf,0,gnc)
P4

Figure 3: T22. Fitting a box with lx<slx and ly<sly

Node P1 holds the tokens associated to boxes while P2 holds the tokens representing the free areas in the pallet. Thus, in transition T22 (Figure 3) two tokens are generated to describe the two new free squares generated due to the space fragmentation.

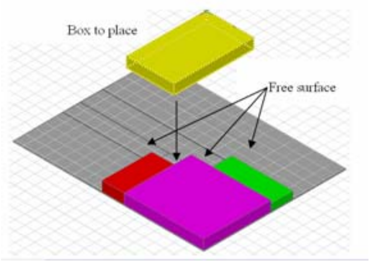


Figure 4: T31. Fitting a box with lx>slx and ly>sly

When a box does not fit in a free area, as shown in Figure 4, but there is a way to place it by compacting different free areas that can be attached altogether, the box is virtually decomposed as a sequence of smallest boxes (virtual box concept). A virtual box is the division of a box into small boxes which must be placed in contiguous free areas, Figure 5.

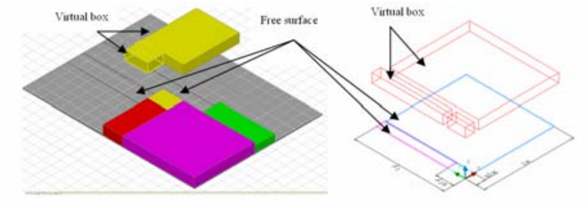


Figure 5: T31. Virtual boxes

Figure 6 illustrates the event that formalizes fitting a box into a free pallet area when the box lengths *x* and *y* are longer than the lengths *x* and *y* of a free pallet surface. Therefore, two new virtual boxes are generated and added in the place *P3* to describe the dimensions. P4 is used to control and coordinate the sequence of events to solve a virtual box placement before any other box could be chosen to be placed in the pallet. Colour *Ge* changes form *0* to *1* in the output arc to mark that the original box has been divided into one or more virtual boxes; the global free surface also is decremented in *sly*slx* which are the dimensions of the free surface used and the number of virtual boxes *gncv* is incremented in two, which are the two virtual boxes generated due to the firing of this transition.

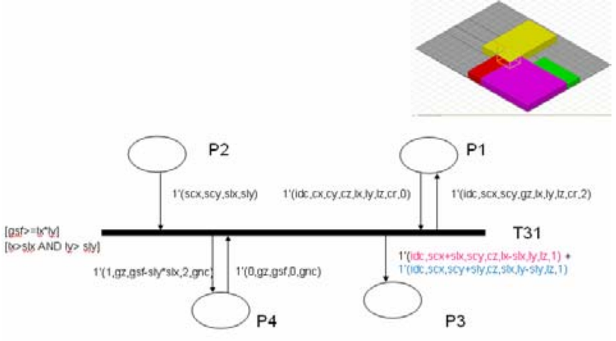

P2
P1
1'(scx,scy,slx,sly)
1'(idc,cx,cy,cz,lx,ly,lz,cr,0)
1'(idc,scx,scy,gz,lx,ly,lz,cr,2)
[gsf>=lx*ly]
[lx>slx AND ly> sly]
T31
1'(1,gz,gsf-sly*slx,2,gnc)
1'(0,gz,gsf,0,gnc)
1'(idc,scx+slx,scy,cz,lx-slx,ly,lz,1) +
1'(idc,scx,scy+sly,cz,slx,ly-sly,lz,1)
P4
P3

Figure 6: T31. Fitting a box with lx>slx and ly>sly

To place a virtual box, two different set of events have been considered: the first set represents the placement of a box that do not overcome a free surface, while the second set of events places a box which does overcome the surface and generate more virtual boxes.

Figure 4 which was used to introduce the concept of virtual box is used again to illustrate the idea of placing virtual boxes. It is assumed that part of the original box has been placed and it remains only one virtual box to be placed as shown in Figure 7.

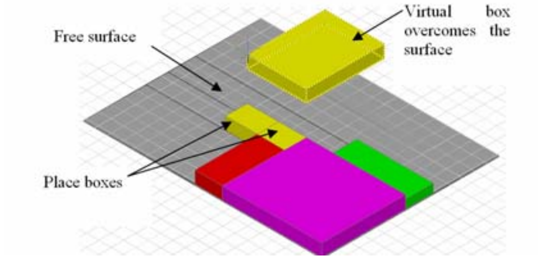


Figure 7: Fitting a box with lx>slx, ly<sly, scx=cx and scy=cy.

In this situation, transition T58 would be fired (see Figure 8), in which a virtual box is placed generating one virtual box, which is indicated by the output arc of place *P3*. Also another free surface is generated, output arc of place *P2*. Finally as explained in the transition *T31*, global free area is decremented and number of virtual boxes keeps steady because one virtual box is placed but another y is created.

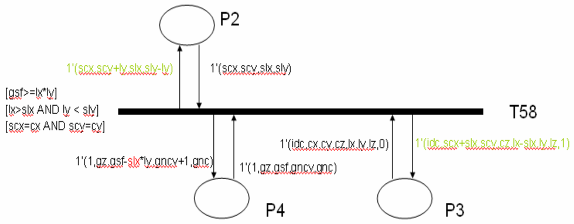

P2
1'(scx,scy+ly,slx,sly-ly)
1'(scx,scy,slx,sly)
[gsf>=lx*ly]
[lx>slx AND ly < sly]
[scx=cx AND scy=cy]
T58
1'(idc,cx,cy,cz,lx,ly,lz,0)
1'(idc,scx+slx,scy,cz,lx-slx,ly,lz,1)
1'(1,gz,gsf-slx*ly,gncv+1,gnc)
1'(1,gz,gsf,gncv,gnc)
P4
P3

Figure 8: T58. Fitting a box with *lx>slx*, *ly<sly*, *scx=cx* and *scy=cy*.

Additionally to all these events that specify how to place a box into a free pallet area, there is a particular event that allows changing the orientation of the box in order to fit better in a free pallet area. This new event can be fired at any time but only once per each box. Figure 9 illustrates the arc expressions that describe this event.

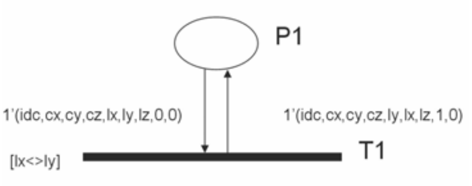


Figure 9: Rotation of a box to be fitted in the pallet

3.3. Coverability tree

The coverability tree relies on the computation of all reachable states and state changes of the system, and it is based on an explicit state enumeration.

By means of the coverability tree, behavioural properties of the system can be verified. Examples of such properties are the absence of deadlocks in the system, the possibility of always reaching a given state, and the guaranteed delivery of a given service. The coverability tree of CP-nets can also be applied to timed CP-nets. Hence, it is also possible to investigate the functional correctness of the proposed model when it is used as a DSS at the final part of a production system.

By using a CPN simulator that can support the evaluation of the coverability tree of the system described under different work loads (different boxes specified in place *P1*) it is possible to check the different combination in which boxes can be fitted in the pallet, and choose the one that minimizes the number of levels of boxes in the pallet.

The specification of the final state consists to force all the tokens in node *P1* to set the colour ce with value 2, mathematically represented by the vector:

$$Mf=[*'(*,*,*,*,*,*,*,2),*,*,*] \qquad (1)$$

Thus, marking M_f can be interpreted that any state with all the tokens in place *P1* with colour *ce=2* can be considered the goal state since all the boxes has been fitted inside the pallet area.

3.4. Heuristics

By considering that the exploration of the whole coverability tree is quite expensive in terms of computer memory requirements and computational time, some heuristics have been designed to avoid the evaluation of certain sequence of events that will not lead a good solution. In Piera (2007) the main aspects of the CPN tool used to support heuristics and knowledge representation to improve the analysis of the coverability tree is presented. This tool has been used to get feasible results solving the pallet loading problem using a reduced number of different types by means of formalizing specific knowledge in terms of heuristics.

3.4.1. Place boxes with the same characteristics

An heuristic has been developed to implement in the model so boxes are placed by groups of the same type. Thus, if a box of certain type is chosen, the model is forced to place all boxes of this type before any other type of box could be chosen.

This heuristic is specified by the introduction of two more transitions to the model, two places and two colours. These colours are *tp*, type box which can take values of *0* if the type of box has never been picked and *1* if it has been picked The other colour added to the model is *nc*, the number of boxes of the same type.

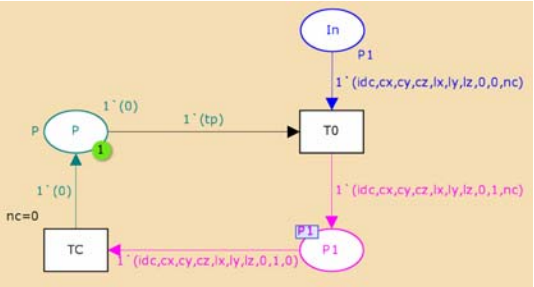

In
P1
1`(idc,cx,cy,cz,lx,ly,lz,0,0,nc)
1`(0)
P
P
1
1`(tp)
T0
1`(0)
nc=0
1`(idc,cx,cy,cz,lx,ly,lz,0,1,nc)
TC
P1
P1
1`(idc,cx,cy,cz,lx,ly,lz,0,1,0)

Figure 10: Introduction of the heuristic

Figure 10 illustrates the changes in the model introduced by the heuristic. Transition *T0* marks a type of box as used by changing *tp* from *0* to *1* while *TC* release this marking only in case all boxes of this type have been place.

4. RESULTS

Several tests to explore the coverability tree have been made based in the CPN tools environment, (CPN Tools home page). CPN tools is a graphical computer tool supporting CP-nets. It consists of three closely integrated components. The CPN editor supports the construction and editing of hierarchical CPN models. Simulation of CPN models is supported by the CPN simulator, while the CPN state space tool supports the state space method of CP-nets. In addition to these three main components, the tool includes a number of additional packages and libraries.

It is possible to attach a piece of sequential CPNML code to obtain the information required.

4.1.1. Scenario evaluations

In this test, seven boxes of two types have been introduced. The initial markings of the boxe place is 3*`(1,0,0,0,40,50,10,0,0)+ +4`(1,0,0,0,20,50,10,0,0).* These boxes must be fitted in a surface of 100x100 units.

Eight feasible solutions were reached, but some of them take a longer way to reach the result, it means more transitions must be fired to reach the final state. Thus, two feasible solutions with the shortest path (number of events used) has been chosen (see table 3, figure 11)).

Table 3: Optimal solutions

Solution No. 1
1`(1,0,0,0,40,50,10,0,2)+1`(1,0,50,0,40,50,10,0,2)+ 1`(1,40,0,0,20,50,10,0,2)+1`(1,40,50,0,20,50,10,0,2) 1`(1,60,0,0,40,50,10,0,2)+1`(1,60,50,0,20,50,10,0,2) 1`(1,80,50,0,20,50,10,0,2)
No empty space
Solution No. 2
1`(1,0,0,0,40,50,10,0,2)+1`(1,0,50,0,40,50,10,0,2)+ 1`(1,40,0,0,40,50,10,0,2)+1`(1,40,50,0,20,50,10,0,2) 1`(1,60,50,0,20,50,10,0,2)+1`(1,80,0,0,20,50,10,0,2) 1`(1,80,50,0,20,50,10,0,2)
No empty space

As it is shown, these feasible solutions place the eight boxes into the pallet leaving no space unused. All necessary information such as coordinates and dimensions of each box is generated and can be checked by evaluating tokens' colours in place *P2*.

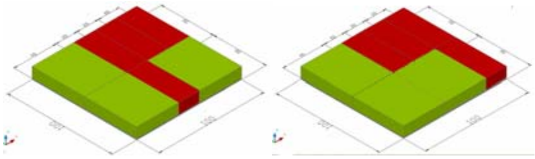


Figure 11: Two feasible solutions

4.1.2. Test results using heuristics

In this test, eight boxes of three different types were introduced. The initial markings of the box place is: *1`(1,0,0,0,60,40,10,0,0,2) ++ 1`(2,0,0,0,40,35,10,0,0,2) ++ 1`(3,0,0,0,30,20,10,0,0,4..*

Again, the coverability tree was explored and two feasible solutions were reached, and both fired the same number of transitions to reach the final state, the information is shown in table 4, (see figure 12).

Using this information it can noticed that the 8 boxes are placed successfully inside the pallet and there is no free space unused. This means that a 100% pallet utilisation can be obtained with all boxes properly placed.

Table 4: Optimal solutions

Solution No. 1
1`(1,40,0,0,60,40,10,0)+1`(1,40,40,0,60,40,10,0)+ 1`(2,0,0,0,40,35,10,0)+1`(2,0,35,0,40,35,10,0)+ 1`(3,0,70,0,20,30,10,1)+1`(3,20,70,0,20,30,10,1)+ 1`(3,40,80,0,30,20,10,0)+1`(3,70,80,0,30,20,10,0)
No empty space
Solution No. 2
1`(1,0,0,0,60,40,10,0)+1`(1,0,40,0,60,40,10,0)+ 1`(2,60,0,0,40,35,10,0)+1`(2,60,35,0,40,35,10,0)+ 1`(3,0,80,0,30,20,10,0)+1`(3,30,80,0,30,20,10,0)+ 1`(3,60,70,0,20,30,10,1)+1`(3,80,70,0,20,30,10,1)
No empty space

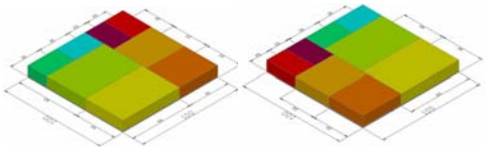


Figure 12: Two feasible solutions

5. CONCLUSIONS AND FURTHER WORK

A pallet maker model approach using the Coloured Petri Net formalism has been presented. The model has been evaluated under different work load conditions, obtaining better results than using commercial software tools as far as pallet utilization it concerns. To improve computational time, some heuristics have been introduced, in such a way that feasible solutions can be obtained.

Due to some shortages inherent to CPN tools, the proposed model has been codified under another logistic focussed CPN environment (see Piera 2004), in order to integrate the model in the DSS of picking process operation that appears in most of production industries.

REFERENCES

CPN Tools home page: http://www.daimi.au.dk/CPNTools/.

Bischoff, E., Dowsland, W.B., 1982, An application of the microcomputer to product design and distribution, *Operational Research Society Journal*.

Chua, C.K., Narayanan, V., Loh, J., 1998 Constraint-based spatial representation technique for the container packing problem, *Integrated Manufacturing Systems*.

Gehring, H., Menschner, K., Meyer, M., 1990, A computer-based heuristic for packing pooled shipment containers, *European Journal of Operational Research*.

George, J.A., Robinson, B.F., 1980, A heuristic for packing boxes into a container, *Computer and Operational Research*, 7, pp. 147–156.

Han, C.P., Knott, K., Egbelu, J.P., 1989, A heuristic approach to the three-dimensional cargo-loading problem, *Int. J. Prod. Res*., 27 (5), pp. 757–774.

Ivancic, N., Mathur, K., Mohanty, B.B.,1989, An integer programming based heuristic approach to the three dimensional packing problem, *Journal of Manufacturing and Operations Management*.

Jensen, K, 1997, *Coloured Petri Nets: Basic Concepts, Analysis Methods and Practical Use*, 2, Springer-Verlag. Berlin.

Kristensen, M.L., 1998, Christensen, S., Jensen, K., The practitioner's guide to coloured Petri nets, *Springer-Verlag*, pp. 98-132.

Lim, A., Rodrigues, B., Yang, Y., 2005, 3-D Container Packing Heuristics *Applied Intelligence, Springer Science + Business Media*, l22, pp. 125–134.

Ngoi, B.K.A., Tay, M.L., Chua, E.S., 1994, Applying spatial representation techniques to the container packing problem, *Int. J. Prod. Res*.

Piera, M. A., et al. 2004, Optimization of Logistic and Manufacturing Systems through Simulation: A Colored Petri Net-Based Methodology, SIMULATION, *Transactions of The Society for Modeling and Simulation International*, 8, pp. 121-129.

Piera, M.A., Mujica M.A. Guasch, 2007, An efficient CPN modeling approach to tackle the pallet packing problem, *in Proceedings of the 6th EUROSIM Congress on Modeling and Simulation*, September 9-13, Ljubljana (Slovenia).

Xue, J., Lai, K.K., 1997, Effective methods for a container packing operation, *Mathl. Comput. Modelling*.

SHOP ORDERS SCHEDULING: DISPATCHING RULES AND GENETIC ALGORITHMS BASED APPROACHES

Antonio Cimino[(a)], Francesco Longo[(b)], Giovanni Mirabelli[(c)], Enrico Papoff[(d)]

[(a) (b) (c) (d)]Modeling & Simulation Center - Laboratory of Enterprise Solutions (MSC – LES)
M&S Net Center at Department of Mechanical Engineering
University of Calabria
Via Pietro Bucci, Rende, 87036, ITALY

[(a) (b) (c) (d)]{acimino, f.longo, g.mirabelli, e.papoff}@unical.it

ABSTRACT

In the wide context of production planning a critical role is played by the operative programming, or short period production planning, whose results affect considerably the production system performances. The research work presented in this paper is focused on the Shop Orders scheduling problem into a real manufacturing system using dispatching rules and genetic algorithms based approaches supported by Modelling & Simulation. The objective is to verify the behaviour of different dispatching rules as well as to test the potentialities of production planning guidelines obtained by using genetic algorithms.

Keywords: Manufacturing Systems, Production Planning, Shop Order Scheduling, Modeling & Simulation, Genetic Algorithms

1. INTRODUCTION

As well known the Shop Orders (S.O.s) scheduling problem within manufacturing systems is usually characterized by high complexity due to the different interacting variables and to the stochastic nature of the system itself (i.e. stochastic process and set-up times, stochastic lead times, etc.). In addition a real manufacturing process is characterized by a number of peculiarities such as machines unavailability (due to failures) machines duplications, S.Os contemporarily worked on more machines, priority S.Os, limited capacity of the intermediate buffers between machines, significant transportation times. The representation of the mentioned aspects by means of analytical models is an exceeding difficult task. In effect the analytical models representing such type of systems are usually characterized by restrictive assumptions. Note that analytical models characterized by restrictive assumptions allow to gain confidence about the S.Os scheduling problem even if they often fall short of results applicability.

One of the most widely used approach for studying scheduling problems within manufacturing systems is the Modeling & Simulation (M&S) approach that gives the possibility to take into consideration the high complexity of a manufacturing system avoiding restrictive assumptions and transferring on the real system the results obtained by using simulation models. Note that also a simulation model usually contains restrictive assumptions. However such assumptions usually aim at defining the physical and logical boundaries of the simulation model (i.e. modeling the inventory management system in a S.Os scheduling problem may not be necessary). In other words all the assumptions made in a simulation model allow to recreate a model that should be valid in its domain of applicability (the Verification, Validation and Accreditation assess the capability of a simulation model to represent a real system with satisfactory accuracy).

The S.Os scheduling activities within a manufacturing system are usually part of the production planning process. In turn the production planning process schedules all the production activities over different period of time: in the long period the planning aims at evaluating the quantity to be produced for each product and the production resources to be used; in the short period the objective is the optimal scheduling of the S.Os on the available machines.

In this paper we developed a simulation model of a real manufacturing system and we studies the S.Os scheduling problem by using both some classical dispatching rules and the genetic algorithms in order to find out specific scheduling guidelines to be used for improving manufacturing system performances.

Before getting into details of the research work let us give a brief summary of the paper. Section 2 describes the manufacturing process being analyzed in this paper. Section 3 proposes the manufacturing process modeling (simulation model development and simulation model verification, validation). Section 4 presents the simulation results. Finally the last section reports the conclusions and the research activities still on going.

2. THE MANUFACTURING PROCESS

The research work has been done in collaboration with a manufacturing system producing small metallic carpentry structures. Due to the high number of different structures the company top management decided to carry out a study devote to improve the efficiency of the short period production planning. In effect such need comes out from the continuous delays in S.Os completion that, in turn, cause the decrease of the customers' satisfaction level. To well understand all the steps of the research work, it is useful to give a brief description of the manufacturing process.

The manufacturing process has to be regarded as a flow shop system in which each S.O. has the same routing, thus the visiting order of the machines is always the same. The main manufacturing operations are described as follows:

- raw materials preparation (ID 1);
- cutting (ID 2);
- drilling (ID 3);
- welding (ID 4);
- assembly (ID 5);
- sandblast (ID 6);
- painting and drying (ID 7).

During the preparation phase all the materials, needed for each Shop Order are taken from the raw materials warehouse. The first operation of the manufacturing process is the cut performed by using a pantograph supported by laser cutting system and equipped for receiving CAM information (*Computer Aided Manufacturing*) directly from the production planning office. All the metallic components are then drilled in order to create all the holes needed for the assembly process. The main components are welding by using two types of welding technologies: MIG/MAG (Gas Metal Arc Welding) and TIG (Gas Tungsten Arc Welding). Thanks to the assembly process all the components are assembled and form the final metallic carpentry structure. The sandblast operation aims at cleaning the metallic surfaces (by using high speed particles that hit the surfaces) before the painting. Finally painting and drying activities complete the manufacturing process.

3. MODELING THE MANUFACTURING PROCESS

Two types of S.O. can enter the system: normal and priority. Usually normal S.Os are scheduled on a 2-weeks time window (each new S.O. enters in the last position of the 2-weeks queue). On the contrary, a priority S.O. can enter the 2-weeks queue in any position at any time (it depends on the priority level of the S.O.). In other words the system allows the *passing* between jobs. Each S.O. has a finite number *m* of operations, one on each machine and it is allowed to work twice a job on the same machine. All the S.Os entered into the system have to be necessarily completed.

Machines could not be available during the scheduling period because of the failures. Failures have been modeled by using a negative exponential distribution for both the Mean Time To Failure (MTTF, expressing the time between two consecutive machine failures) and the Mean Time To Repair (MTTR, expressing the time required for repairing the machine). Finally process and set-up's time are considered as stochastic variables each one with a specific statistical distribution. According to these hypotheses it follows that the case analysed belong to the *dynamic-stochastic* scheduling problem because new S.Os arrive during the scheduling horizon and most of the numerical quantities are stochastic.

The main steps of the simulation model development can be summarized as follows:

- data collection and distributions fitting;
- simulation model implementation;
- simulation model verification and validation.

3.1. Data Collection and Distribution fitting

The most important information were collected by means of interview and by using the company informative system. Data collected regard bill of materials, S.Os routing, S.Os inter-arrival times, number of S.Os for each customer, inventory control policies and suppliers lead times, process and set-up times, machines downtimes and uptimes, material handling modes and times.

All the stochastic variables have been analyzed in order to find out statistical distributions capable of fitting the empirical data with satisfactory accuracy. Figure 1 shows the histogram and the statistical distribution of the process time of the assembly operation. Figure 2 shows the histogram and the statistical distribution of the process time of the drilling operation.

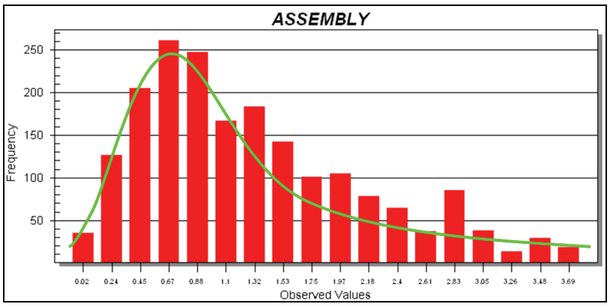


Figure 1: Histogram and Statistical Distribution of the Process Time of the Assembly Operation

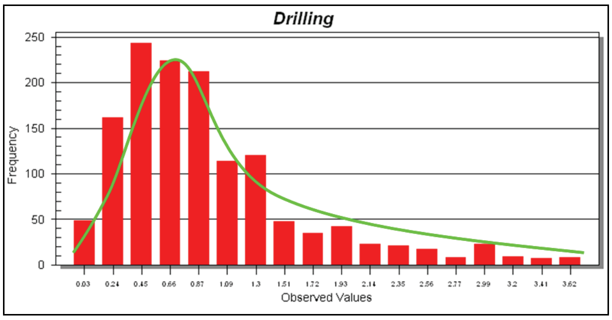


Figure 2: Histogram and Statistical Distribution of the Process Time of the Drilling Operation

3.2. The simulation model development

The simulation model was developed by using the discrete event simulation software eM-Plant by Tecnomatix Technologies. The main idea was to develop a flexible and time efficient simulator. A flexible simulator should be capable of easily integrating additional features as the time goes by; a time efficient simulator should require few time for executing simulation run.

Note that the simulator flexibility cannot be easily achieved if library objects are used for developing the simulator architecture. In effect each library object should represent a specific component/part of a real system; sometime such objects do not represent the real system with satisfactory accuracy. The solution to this problem is the simulator development by using programming code. eM-Plant provide the users with a simulation language (Simple++) that can be used for implementing classes and objects. Such classes can be accessed and modified at any time (also saved and used in other simulation models) assuring, as a consequence, high level of flexibility in terms of both model accuracy and future changes.

Concerning the computational efficiency of the simulator and the time required for executing simulation runs, we should take into consideration how a discrete event simulation software works. In a discrete event system the state of the system changes at discrete event time points due to the flow of entities inside the system (i.e. end of an operation, arrival of a new shop order, etc.). In other words entities take actions that change the state of the system. Usually entities are defined as classes instantiated inside the simulation model. Each entity can also have attributes used for storing specific information. Note that the higher is the number of entities flowing in the simulation model the higher is the computational load of the simulator. Consider the case of a manufacturing process in which thousands of components and products usually flow inside the system (it means thousands of entities flowing inside the simulation model). The approach used for developing the simulation model proposed in this paper is based on the idea to substitute the flow of entities with a flow of information opportunely stored in tables. The events generation is committed to specific objects (provided by the eM-Plant library) called event generators.

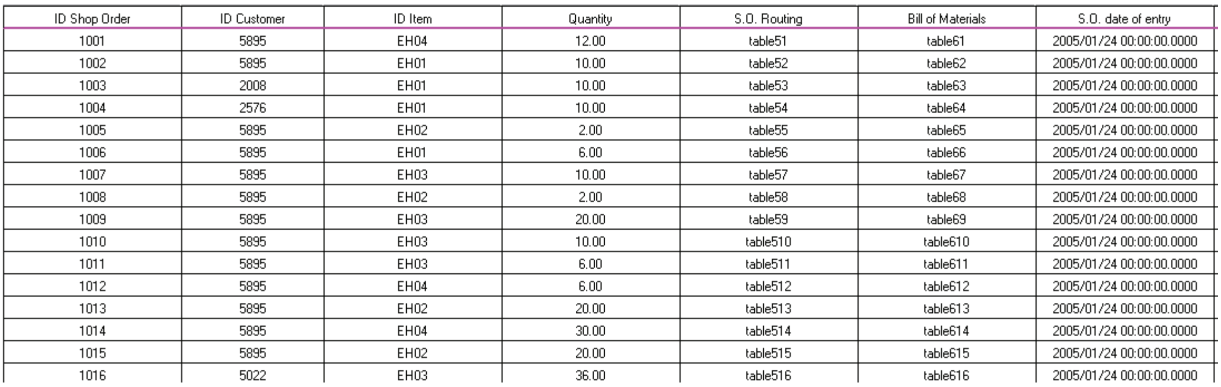
The change of the state of the system, due to the generation of an event, is managed by ad-hoc programmed routines; the programming code also takes care of updating all the information stored in the tables.

By following this approach, two main advantages can be obtained: (i) a great gain in term of computational load of the simulator; (ii) reduction of the time required for executing simulation runs. Figure 3 shows an example of information stored in table for each entity (shop order) flowing into the simulator.

The simulator main frame is called *model*. It contains 9 secondary frames. Each frame is built to recreate a specific operation of the real manufacturing system. In particular 7 frames recreate the operations described in section 2 (raw materials preparation, cutting, drilling, welding, assembly, sandblast, painting and drying) whilst the remaining 2 frames are respectively the *Production Manager* (PM) and the *Graphic User Interface* (GUI). The PM generates the S.Os and the relative production planning, takes care of S.Os scheduling, resource allocation and inventory management. The graphic user interface provides the user with many commands as, for instance, simulation run length, start, stop and reset buttons and a Boolean control for the random number generator (to reproduce the same experiment conditions in correspondence of different operative scenarios). Furthermore the GUI allows the user to select the dispatching rule to be used for S.Os scheduling or to select S.Os scheduling based on the results of genetic algorithms.

Let us introduce now the performance indexes implemented in the simulation model used for evaluating the goodness of the S.Os scheduling. We propose a multi measure approach based on orders completion time and on due dates. In particular the simulator monitors for each S.O. the following performance measures: the average and the variance of the *Flow Time* (FT), the average and the variance of the *Lateness* (LT) and the *Fill Rate* (FR). The FT of the *i-th* S.O. is the difference between the S.O. Completion Time (CT) and the S.O. Release Time (RT) as reported in equation 1.

$$FT_i = CT_i - RT_i \tag{1}$$



ID Shop Order	ID Customer	ID Item	Quantity	S.O. Routing	Bill of Materials	S.O. date of entry
1001	5895	EH04	12.00	table51	table61	2005/01/24 00:00:00.0000
1002	5895	EH01	10.00	table52	table62	2005/01/24 00:00:00.0000
1003	2008	EH01	10.00	table53	table63	2005/01/24 00:00:00.0000
1004	2576	EH01	10.00	table54	table64	2005/01/24 00:00:00.0000
1005	5895	EH02	2.00	table55	table65	2005/01/24 00:00:00.0000
1006	5895	EH01	6.00	table56	table66	2005/01/24 00:00:00.0000
1007	5895	EH03	10.00	table57	table67	2005/01/24 00:00:00.0000
1008	5895	EH02	2.00	table58	table68	2005/01/24 00:00:00.0000
1009	5895	EH03	20.00	table59	table69	2005/01/24 00:00:00.0000
1010	5895	EH03	10.00	table510	table610	2005/01/24 00:00:00.0000
1011	5895	EH03	6.00	table511	table611	2005/01/24 00:00:00.0000
1012	5895	EH04	6.00	table512	table612	2005/01/24 00:00:00.0000
1013	5895	EH02	20.00	table513	table613	2005/01/24 00:00:00.0000
1014	5895	EH04	30.00	table514	table614	2005/01/24 00:00:00.0000
1015	5895	EH02	20.00	table515	table615	2005/01/24 00:00:00.0000
1016	5022	EH03	36.00	table516	table616	2005/01/24 00:00:00.0000

Figure 3: An example of information stored in table for each entity (shop order) flowing into the simulator

The LT of the *i-th* S.O. is the difference between the S.O. Completion Time and the S.O. Due Date (DD), as expressed by equation 2.

$$FT_i = CT_i - DD_i \qquad (2)$$

Finally the FR is the percentage of S.Os meeting the due date as expressed by equation 3.

$$FR_i = \frac{\sum_{i=!}^{k} S.O._i}{\sum_{i=1}^{n} S.O._i} \qquad (3)$$

3.3. Simulation model Verification and Validation

The accuracy and the quality throughout a simulation study are assessed by conducting verification and validation processes (Balci 1998). Usually a real world system is abstracted by a conceptual model; in turn a conceptual model is then translated into a computerized simulation model. The verification aims at determining if the computerized simulation is an accurate translation of the initial conceptual model. A simulator must substitute the real system for the purpose of experimentation; to this end the simulator has to represent the real system with satisfactory accuracy. The level of accuracy is usually evaluated by the validation phase. For further details on simulation model Verification & Validation, refer to the American Department of Defence Directive 5000.59.

The simulator verification has been carried out by using the *Assertion Checking* dynamic technique. Such technique aims at checking what is happening inside the simulator against what we assume happening (further information in Adrion et al. 1982). In case of checking discordance, the technique reveals an error usually due to incorrect programming code or values. To detect errors inside the simulator we inserted global, region and local assertion in order to verify the entire model. A number of different errors were identified by the assertions and successively corrected (i.e. errors on S.O. routing, on machines set-up times, on raw materials inventory management, etc.

The simulator validation has been carried out by using the Mean Square Pure Error analysis (MSPE). The MSPE aims at evaluating the length of the simulation run that guarantees the goodness of the statistical results in output from the simulation model.

Considering the stochastic distributions implemented in the simulation model we can assert that the outputs of the simulation model are subjected to an experimental error with normal distribution, N(0, σ^2). The best estimator of σ^2 is the mean squares error. The simulation run has to be long enough to have small values of the MSpE of the performance measures being considered. In other words, the experimental error must not “cover” the simulation results. Considering the Flow Time, we can write:

$$MSpE(t) = \sum_{h=1}^{n} \frac{(FT_h(t) - \overline{FT}(t))}{n-1} \qquad (4)$$

- $FT_h(t)$, value of the Flow Time at instant of time t during the replication h;
- $h=1,\dots,n$ number of replications.

Analogous equation can be written for the LT and the FR. The simulation run length chosen is 200 days. Such time, evaluated with four replications, assures a negligible mean squares error for the Flow Time. The same analysis for the Lateness and the Fill Rate gives lower simulation run lengths.

3.4. Genetic Algorithms implementation to support Shop Order scheduling

Once tested the validity of the simulation model, further implementations were carried out to introduce Genetic Algorithms (GA) as support tool for short period production planning. The GA was implemented as functional part of a particular tool called optimizer. This object aims at:

- optimising S.Os scheduling by means of GA;
- testing the proposed scheduling;
- monitoring the manufacturing system performances by using the Flow Time, the Lateness and the Fill Rate indexes.

It is important to highlight the nature of problem which must be solved by the optimizer and, of course, understand how it works. The problems concerning the stochastic shop orders scheduling cannot be solved only by means of simulation tools. In effect, after establishing a certain S.Os scheduling a simulation model can only evaluates the system performance under the scheduling proposed. By proposing a new S.Os scheduling, the initial solution can be improved or worsened. To improve the S.Os scheduling it is therefore necessary to use optimization algorithms which, thanks to an interface with the simulation model, find out the most suitable solution optimizing the scalar function chosen to measure scheduling goodness (i.e. the Flow Time). Optimization algorithms must find out acceptable solutions, while the simulation model must test, validate and choose the best solutions.

The interface between the simulation model and genetic algorithms was created through the programming of specific sub-routines, written using the simulation language Simple++. The use of genetic algorithms goes through three fundamental steps: (i) initial S.Os scheduling (proposed by the user); (ii) setting of genetic operators and algorithms initialization (iii) optimization.

4. SIMULATION RESULTS AND ANALYSIS

The research work focalizes on the Shop Orders scheduling problem into a real manufacturing system using dispatching rules and genetic algorithms based approaches supported by Modelling & Simulation. The objective is to verify the behaviour of different dispatching rules as well as to test the potentialities of production planning guidelines obtained by using genetic algorithms.

The scheduling rules (implemented in the simulator) being tested in the following analysis are: (i) the Shortest Production Time (SPT); (ii) the Due Date (DD); (iii) the Longest Production Time (LPT).

Table 1 reports the average values of the FT, LT and FR in correspondence of each scheduling rule. The best performance in terms of flow time is guaranteed by the SPT rule, while the best performance in terms of LT and FR is guaranteed by the DD rule. Table 2 reports the standard deviation values for each performance measure in correspondence of each scheduling rule.

Table 1: Shop Orders Scheduling Rules and average values of the Performance Measures

	Flow Time (FT) [days]	**Lateness (LT) [days]**	**Fill Rate (FR) [%]**
SPT	4.1	1.7	87.38
DD	4.8	1.2	90.40
LPT	6.4	2.6	83.26

Table 2: Shop Orders Scheduling Rules and standard deviation of the Performance Measures

	Flow Time (FT) [days]	**Lateness (LT) [days]**	**Fill Rate (FR) [%]**
SPT	0.032	0.030	0.230
DD	0.041	0.033	0.190
LPT	0.037	0.036	0.210

Figure 4 shows the FT and the LT versus the scheduling rules. Note that the DD performs better in terms of respect of the due dates.

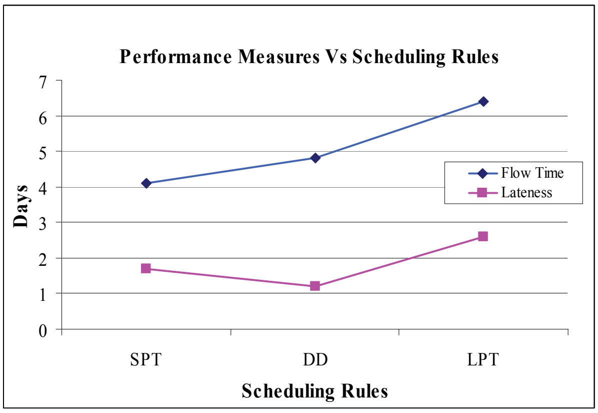


Figure 4: Performance Measures Vs Scheduling Rules

A number of simulation runs have been also made for investigating the S.Os scheduling by using the genetic algorithms. Three different optimizations have been carried out respectively trying to minimize the FT, minimize the LT and maximize the FR. Table 3 reports the simulated FT in correspondence of each generation; in particular for each generation the best value, the average value and the worst value are reported.

Table 3: Flow Time Optimization: Best Average and Worst Solutions found by GA

Generation	**FT Best**	**FT Average**	**FT Worst**
1	9.00	10.00	10.60
2	7.50	8.30	9.60
3	6.60	7.50	9.00
4	6.00	7.00	8.80
5	5.70	6.80	8.30
6	5.40	6.50	7.90
7	5.30	6.20	7.60
8	5.00	6.00	7.40
9	4.90	5.70	7.00
10	4.85	5.40	6.50
11	4.70	5.30	6.20
12	4.50	5.10	5.80
13	4.30	4.90	5.50
14	4.20	4.70	5.00
15	4.00	4.20	4.50
16	3.90	4.00	4.30
17	3.85	3.90	4.00
18	3.75	3.80	3.90
19	3.75	3.80	3.80
20	3.70	3.80	3.80
21	3.70	3.70	3.70
22	3.70	3.70	3.70
23	3.70	3.70	3.70

After 23 replications the best, the average and the worst solutions converge to the value 3.70 days. Note that such value is lower than best result obtained with the SPT rule (the improvement is about 9.8%). The figure 5 reports the performance graph that shows the FT optimization

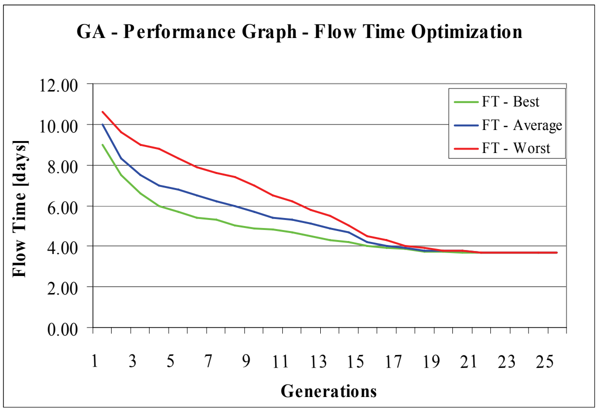


Figure 5: Flow Time Optimization

Analogously the table 4 reports the optimization results for the LT (best, average and worst values over 23 generations).

Table 4: Lateness Optimization: Best Average and Worst Solutions found by GA

Generation	LT Best	LT Average	LT Worst
1	4.25	5.05	6.05
2	3.75	4.65	5.45
3	3.25	4.35	5.05
4	3.00	3.95	4.85
5	2.85	3.75	4.70
6	2.65	3.50	4.45
7	2.50	3.30	4.25
8	2.25	3.05	4.00
9	2.00	2.85	3.70
10	1.85	2.75	3.50
11	1.75	2.60	3.25
12	1.65	2.35	3.00
13	1.50	2.05	2.90
14	1.40	1.85	2.65
15	1.30	1.60	2.35
16	1.30	1.40	2.05
17	1.20	1.30	1.75
18	1.15	1.25	1.50
19	1.10	1.15	1.40
20	1.10	1.15	1.30
21	1.05	1.10	1.20
22	1.05	1.10	1.20
23	1.05	1.05	1.10

After 23 replications the best, the average and the worst solutions converge to the value 1.05 days. Note that such value is lower than best result obtained with the DD rule (the improvement is about 12.5%). The figure 6 reports the performance graph that shows the LT optimization

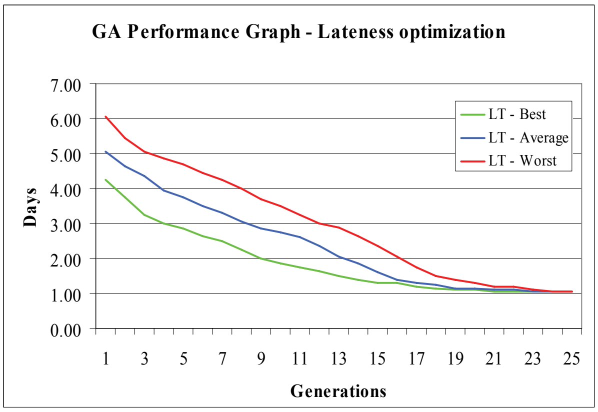


Figure 6: Lateness Optimization

Finally the table 5 reports the optimization results for the FR (best, average and worst values over 23 generations). After 23 replications the best, the average and the worst solutions converge to the value 95%. Note that such value is greater than the best result obtained with the DD rule (the improvement is about 4.6%). The figure 7 reports the performance graph that shows the FR optimization.

Table 5: Fill Rate Optimization: Best Average and Worst Solutions found by GA

Generation	FR Best	FR Average	FR Worst
1	81.25	79.00	78.00
2	82.33	80.70	79.10
3	83.10	81.50	80.20
4	84.20	82.25	81.00
5	85.00	83.30	81.90
6	85.80	84.40	82.70
7	86.70	85.50	83.90
8	87.90	86.50	84.10
9	89.00	87.70	85.50
10	90.05	89.00	86.70
11	90.88	89.65	87.90
12	91.56	90.32	89.00
13	92.21	91.15	89.90
14	92.78	91.99	90.50
15	93.50	92.10	91.40
16	93.75	93.00	92.10
17	94.00	93.50	92.80
18	94.25	93.80	93.20
19	94.35	94.20	93.80
20	94.50	94.40	94.15
21	94.77	94.60	94.45
22	95.00	94.90	94.80
23	95.00	95.00	95.00

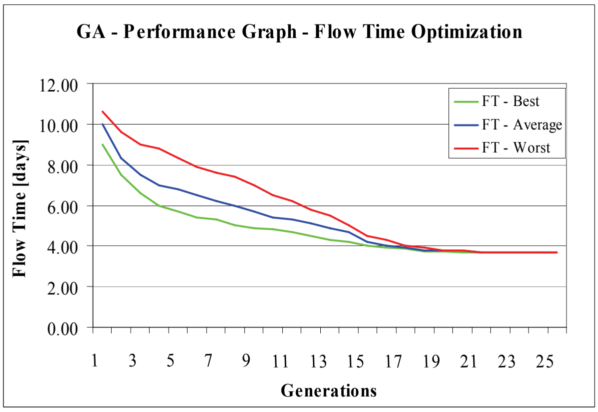


Figure 7: Fill Rate Performance Graph

5. CONCLUSIONS

The main goal of the research study was to verify the behaviour of different dispatching rules and the potential of genetic algorithms for the S.Os scheduling within a manufacturing system devoted to produce metallic carpentry structure. To this end the authors implemented a discrete simulation model by using an advanced modeling approach.

The analysis carried out show the behavior of three different scheduling rules in terms of Flow Time, Latiness and Fill Rate. In addition, three different optimizations have been made on the FT, LT and FR by using the genetic algorithms. The genetic algorithms are capable of finding better shop orders scheduling improving the results obtained by using the classical scheduling rule.

REFERENCES

Aytug, H., Khouja, M., Vergara, F.E., 2003. *Use of genetic algorithms to solve production and operations management problems: A review*, International Journal of Production Research 41 (17), 3955–4009.

Balci, O., 1998. *Verification, Validation and Testing*. In Handbook of Simulation, edited by J. Banks, pp. 335-393, New York: Wiley Interscience

Banks, J., 1998. *Handbook of simulation*, New York: Wiley Interscience.

Brah, S. A., Hunsucker, J. L., 1991. *Branch and bound algorithm for the flow shop with multiple processors*. European Journal of Operational Research, 51, 88–99.

Bruzzone, A.G., Bocca, E., Longo, F., Massei, M., Logistics Node Design and Control over the Whole Life Cycle based on Web Based Simulation, *International Journal of Internet Manufacturing and Services*, I(1), pp. 32-50.

Bruzzone, A.G., Longo, F., Brandolini, M., 2006. Enhancement Process Based on Simulation Supporting Efficiency & Organization, *Proceedings of Summer Computer Simulation Conference*, July 30th – August 03rd, Calgary Canada.

Curcio, D., Longo, F., Mirabelli, G., 2007. Manufacturing Process Management Using A Flexible Modeling & Simulation Approach. *Proceedings of Winter Simulation Conference*, 9-12 December, Washington D.C., USA.

De Sensi, G., Longo, F., Mirabelli, G., Inventory policies analysis under demand patterns and lead times constraints in a real supply chain, *International Journal of Production Research,* doi: 10.1080/00207540701528776

Demirkol, E., Uszoy, R., 2000. *Decomposition methods for reentrant flow shops with sequence dependent setup-times*, Journal of Scheduling 3, 155–177.

Department of Defense, Deputy under Secretary of Defence, DoD modelling and simulation (M&S) management, 1994. DoD Directive 5000.59.

Jones, A., Rabelo, L., Yih, Y., 1995. *A hybrid approach for real-time sequencing and scheduling.* International Journal of Computer Integrated Manufacturing, 8(2), 145–154.

Longo, F., Mirabelli, G., An Advanced Supply Chain Management Tool Based on Modeling & Simulation, *Computer and Industrial Engineering*, 54(3), pp 570-588.

Longo, F., Mirabelli, G., Papoff, E., 2006. Material Flow Analysis and Plant Lay-Out Optimization of a Manufacturing System, *International Journal of Computing*, 5(1), pp. 107-116.

Longo, F., Mirabelli, G., Papoff, E., 2006. Modeling, Analysis & Simulation of Tubes Manufacturing Process and Industrial Operations Controls, *Proceedings of Summer Computer Simulation Conference*, July 30th – August 03rd, Calgary Canada.

Monch, L., Rose, O., Sturm, R., 2003. *A simulation framework for performance assessment of shop-.oor control systems*. Simulation: Transactions of the Society of Modeling and Computer Simulation International 79 (3), 163–170.

Montgomery, D. C., Runger, G. C., 2006. *Applied Statistics and Probability for Engineers*. New York: Wiley Edition.

Park BJ, Choi HR, Kim HS, 2003. *A hybrid genetic algorithm for the job shop scheduling problems*. Computer and Industrial Engineering 45, 597–613.

Pinedo, M.L., 2002. *Scheduling: Theory, Algorithms, and Systems*, second ed., Englewood Cliffs: Prentice-Hall.

Sim, S.K., Yeo, K.T., Lee, W.H., 1994. *An expert neural network system for dynamic job shop scheduling*. International Journal of Production Research, 32(8), 1759–1773.

AUTHORS BIOGRAPHY

ANTONIO CIMINO was born in Catanzaro (Italy) in October the 1th, 1983. He took his degree in Management Engineering, summa cum Laude, in September 2007 from the University of Calabria. He is currently PhD student at the Mechanical Department of University of Calabria. His research activities concern the integration of ergonomic standards, work measurement, artificial intelligence and Modeling & Simulation tools for the effective workplace design.

FRANCESCO LONGO took the degree in Mechanical Engineering from University of Calabria (2002) and the PhD in Industrial Engineering (2005). He is currently researcher at the Mechanical Department (Industrial Engineering Section) of University of Calabria. His research interests regard modeling & simulation of manufacturing systems and supply chain management, vulnerability and resilience, DOE, ANOVA. He is Responsible of the Modeling & Simulation Center – Laboratory of Enterprise Solutions (MSC-LES).

GIOVANNI MIRABELLI was born in Rende in 1963 and he took the degree in Industrial Engineering at the University of Calabria. He is currently researcher at the Mechanical Department of University of Calabria. His research interests include ergonomics, methods and time measurement in manufacturing systems, production systems maintenance and reliability, quality.

ENRICO PAPOFF was born in Naples (Italy) on February the 03rd, 1948. He took the degree in Mechanical Engineering from University of Napoli Federico II, in 1973. He is currently Associate Professor at the Mechanical Department (Industrial Engineering Section) of the University of Calabria. His research interests regard project management and business plans.

SENSITIVITY ANALYSIS AND OPTIMIZATION OF DIFFERENT INVENTORY CONTROL POLICIES ALONG THE SUPPLY CHAIN

Duilio Curcio[(a)], Francesco Longo[(b)]

[(a) (b)] Modeling & Simulation Center - Laboratory of Enterprise Solutions (MSC – LES)
M&S Net Center at Department of Mechanical Engineering
University of Calabria
Via Pietro Bucci, Rende, 87036, ITALY

[(a) (b)] {dcurcio, f.longo}@unical.it

ABSTRACT

This paper focuses on the inventory management problem along the supply chain. A three three-echelons supply chain made up of suppliers, distribution centers and retail stores is considered. The analysis of multiple inventory control policies monitored by using multiple performance measures is proposed. To this end a simulation model, capable of recreating the complex supply chain environment, has been developed.

Keywords: sensitivity analysis, inventory control, supply chain

1. INTRODUCTION

According to Lee and Billington (1993), a Supply Chain (*SC*) is a network of different entities or nodes (plants, distribution centers, warehouses and retailers) which provides material, transform them in intermediate or finished products and deliver them to customers in order to satisfy market requests.

Each SC node is identified by two different parameters:

- the demand;
- the productive capacity.

In order to define each parameter a great amount of data have to be collected. Moreover, information and material flow management among SC nodes becomes a very complex task characterized by a number of critical issues related, for example, to demand (volume and production range), processes (machines downtimes, transportation modes), and supply (parts quality, delivery schedules). The Supply Chain Management (SCM) takes care of the above mentioned issues, studying and optimizing the flow of materials, information and finances along the entire supply chain.
The main goal of a supply chain manager is to guarantee the correct flows of goods and information throughout the SC nodes for assuring the right goods be delivered in the right place and quantity at the right time. Among the others, the inventory management problem along the supply chain plays a critical role in terms of supply chain performances. Lee and Billington (1993) consider the inventory control as the only tool to protect SC stability and robustness. In effect, the objective of the Supply Chain Inventory Management (*SCIM*) is to satisfy the ultimate customer demand increasing the quality and service level and decreasing at the same time total costs; inventories affect SC costs and performance in terms of:

- values tied up, e.g. raw materials have a lower value than finished products;
- degrees of flexibility, e.g. raw materials have higher flexibility than the finished products because they can be easily adopted for different production process;
- levels of responsiveness, e.g. products delivery could be made without strict lead times whereas raw materials transformation usually requires stringent lead times.

During the last years a number of research studies on SCIM have been proposed. Minner (2003) proposes a review on Inventory Models (*IMs*) and addresses their contribution to SC performance analysis. In particular, models analyzed concern to:

- different SC configurations (single/multi-echelon systems): Dellaert and De Kok (2004) present an integrated approach for resource and production management of an assembly system; Chen and Lee (2004) implement an analytical model for demand variability, delivery modes, inventory level and total costs in a multi-echelon SC network;
- parameters variability, i.e. demand disruptions: see Qi et al. (2004) who analyze deviation costs of a one supplier – one retailer after demand disruption; order quantity as reported in Zhou et al. (2007) who introduce an algorithm to compute the parameters of a single item-periodic review inventory policy;

- constraints, i.e. De Sensi et al. (2007) propose the analysis of different inventory control policies under demand patterns and lead time constraints in a real supply chain; Longo and Mirabelli (2008) analyze the effects of inventory control policies, lead times, customers' demand intensity and variability in three different supply chain performance measures. Chen and Krass (2001) propose a new inventory approach, based on the minimal service level constraint which consists in achieving a minimum defined service level in each period; Huang et al. (2005) study the impact of the delivery mode on a one-warehouse multi-retailer system, in order to evaluate the optimal inventory ordering time and the economic lot size for reducing total inventory costs. Inderfurth and Minner (1998) investigate an analytical model to determine safety stocks considering as constraint different service levels.

Analytical models for inventory management take into account all the parameters which affect the inventory level. Allen and D'Esopo (1968) in their research work propose a review of the re-order point order-quantity policy introducing a new time parameter, the expedited leadtime, lower than the normal procurement leadtime. Ramasesh et al. (1991) propose a variant of the same policy with parameters related to demand variability and lead times in order to minimize total purchasing, delivery and inventory costs.

During the years one of the most important tool for studying inventory management along the supply chain has been the Modeling & Simulation based approach. In effect, the evaluation of the performance of different entities involved in the SC, from suppliers to final customers passing through distribution centers, considering a number of stochastic variable and parameters is a quite complex task in which analytical models often fall short of results applicability.

Bhaskaran (1998) carries out a simulation analysis of SC instability and inventory related to a manufacturing plant: in this case, simulation is used to better understand the effects of different inventory strategies on the SC structure. In this context, artificial intelligence techniques (i.e. fuzzy theory and genetic algorithms) combined with the simulation models support the decision making process.

Gupta et al. (2007) apply the genetic algorithm theory for investigating an inventory model of a system characterized by a single item with undefined inventory costs under the effect of different marketing strategies. Giannoccaro and Pontrandolfo (2002) propose an artificial intelligence algorithm in order to manage inventory decisions at all SC stages (optimizing the performance of the global SC). Huang et al.(2005) solve the ordering and positioning retailer inventories problem at the warehouse and stores, satisfying specific customer demand and minimizing total costs by using neural network approaches. Long et al. (2004) propose a revisited Economic Order Quantity (EOQ) model characterized by the introduction of fuzzy lead times.

Different commercial software and programming languages have been used for developing the simulation models. Lee and Wu (2006) model the reorder point order-quantity and the periodic review order-up policy of a distribution system by using the commercial package eM-Plant™; Al-Rifai and Rossetti (2007) adopt Arena™ for testing a new analytical model for a two-echelon inventory system, whereas Bertazzi et al. (2005) implement in C++ a vendor-managed inventory policy in order to minimize purchasing, replenishment and delivery total costs.

2. THE SUPPLY CHAIN CONCEPTUAL MODEL

The SC considered in this research work is made up by three stages as shown in Figure 1:

- *M* manufacturing plants (*MPs*);
- *N* distribution centers (*DCs*);
- *J* stores or retails (*STs*).

Product demand is defined by the final customer.

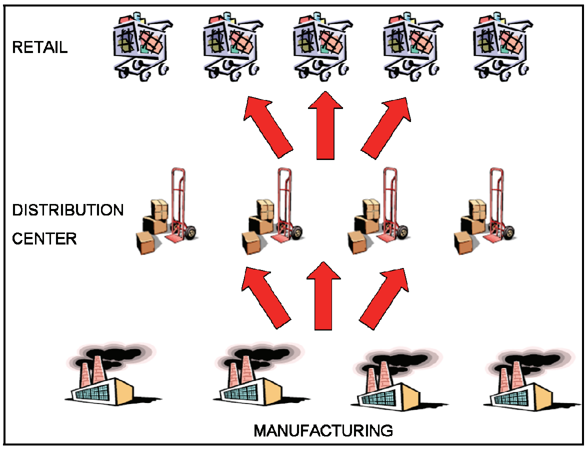


Figure 1: The Conceptual Model of the Supply Chain

2.1. The manufacturing plants

As above mentioned the SC being analyzed has *M* manufacturing plants. Each manufacturing plant has *K* identical processes and it is equipped in order to manufacture *I* different types of products. Variables characterizing the manufacturing plants are:

- the setup time due to plant switching from one product to another;
- the processing time dependent on order size;
- the production capacity.

2.2. The distribution centers

The second SC level is made up by *N* Distribution Centers (*DC*) which store all the *I* products. The inventory control within each DC is as follows. When an order arrives to a DC, the current inventory position (on-hand inventory plus the quantity already on order minus the quantity to be shipped) is checked. If the order is fully satisfied, its status is considered completed and, as a consequence, the inventory level of the DC is decreased of the order quantity, otherwise a lost quantity is recorded. The most important performance measures within a DC are the number of fully satisfied and partially satisfied store orders per period the total lost sales quantity per period to define the service level and the fill rate

2.3. The stores

The third SC level is represented by stores. Each ST works on an eight-hours shift. At the beginning of the day an inventory review is made at each store in order to decide about an order emission at one of the *N* distribution centers. It is necessary to underline that, according to this policy, store orders are delayed until the beginning of the next day so order replenishment is guaranteed by the DC the following day.

Each store chooses the DC capable of replenishing the maximum order quantity requested. The quantity to order for each product is defined after by checking the current on-hand inventory at the store. At each ST, number of fully or partially satisfied orders, lost sales quantity and total quantity ordered for each product is recorded. These data are then used for performance measure evaluation (e.g. the service level, fill rate, etc.).

3. THE SIMULATION MODEL

The goal of this research work is to analyze the supply chain performance implementing three different Inventory policies in order to estimate the inventory level at each SC stage and, as a consequence, inventory costs.

The simulation model is implemented using the commercial simulation software eM-Plant™ by Tecnomatix Technologies. The simulation model development makes use of an advance modeling approach. The classical modeling approach based on library objects to reproduce static and dynamic entities (materials flow, machines, production line, etc.) is replaced with a new one that substitutes flows of entities with a flow of information stored in tables. To access, update and record such information stored in tables, ad-hoc programmed routines have been implemented. The modeling approach proposed has the advantage to allow high flexibility levels in terms if simulation model modification in order to reproduce several system behaviors under different operative environments (further information can be found in Longo & Mirabelli, 2008). Note that the main disadvantage of the modeling approach being used is the animation: generally animation reproduces the entities flow within the simulation model; in this case, the animation is not considered a priority aspect of the simulation study although model structure allows animation implementation.

3.1. The Manufacturing Plants Model Implementation

The part of the simulation model representing the manufacturing plants, recreates the items production process. If all the machines of a plant are busy when an order arrives, this order is queued waiting for another available resource. The switch from an product type to another requires a set up time. No warehouses are available at each plant (make-to-order system).

DCs select plants to send the order on the basis of lead time and quantity that the plant can refurbish. Figure 2 shows the block diagram of the MP selection process.

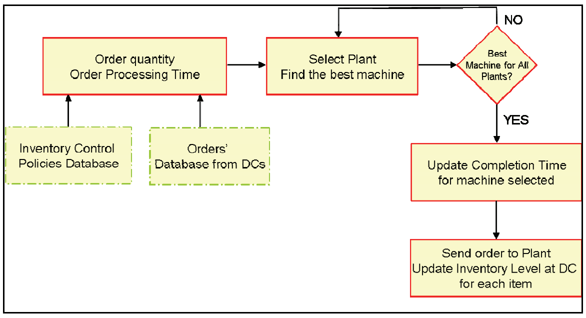

Order quantity
Order Processing Time
Select Plant
Find the best machine
Best Machine for All Plants?
NO
YES
Inventory Control Policies Database
Orders' Database from DCs
Update Completion Time for machine selected
Send order to Plant
Update Inventory Level at DC for each item

Figure 2: Block diagram for the manufacturing plant selection

The simulation model, according to the DCs orders, evaluates total process times and set up times, then selects the best machine in each plant (capable of processing the order) and finally the best machine among all the plants. The selection of the best machine is made according to the time required to complete waiting orders and to start the new order.

3.2. The Distribution Centers Model Implementation

At the beginning of the day the purchase orders from stores arrive at the DCs. The model checks the inventory levels to verify if incoming orders can be satisfied. In each DC, Items number of incoming orders is compared with the on-hand inventory level. If there is enough on hand inventory, STs demand is totally satisfied and the on-hand inventory level is updated; if demand is partially satisfied or unsatisfied, lost sales are recorded.

Each DC emits purchase orders toward the manufacturing plant on the basis of demand forecast. The main activities that take place within each DC are summarized by the block diagram in Figure 3.

3.3. The Stores Model Implementation

The activities performed by the part of the simulation model representing the stores are quite similar to the activities performed in the DCs. In effect at the end of the day the inventory level is checked in order to evaluate whether or not a purchase order has to be emitted.

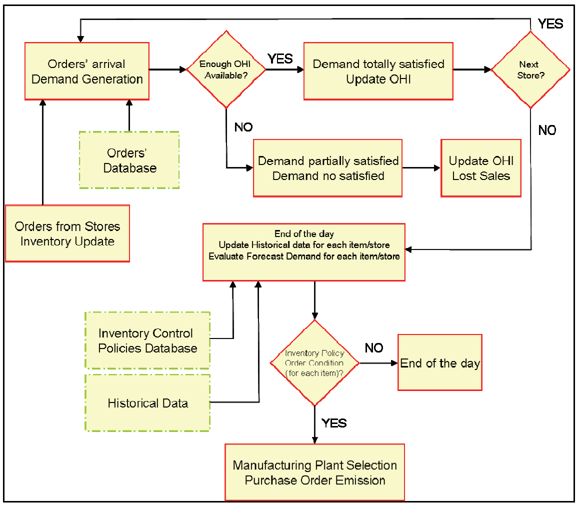


Figure 3: The Distribution Centers Block Diagram

The block diagram in Figure 4 describes the activities performed by the simulation model within each store.

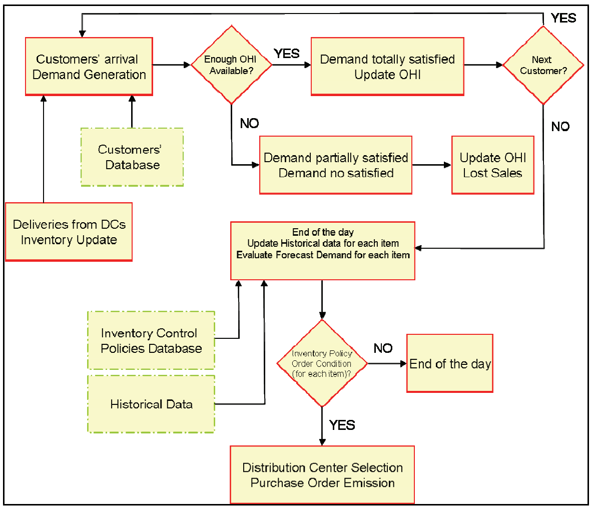


Figure 4: The Stores Block Diagram

4. THE INVENTORY CONTROL POLICIES

In this research work, authors investigate an inventory stocking problem.

In detail, the focus consists in comparing five different Inventory Management Policies (*IMPs*) by using a Modeling & Simulation approach in order to test the performance of the SC analyzed.

According to the SCIM principles, the objective of the inventory control policies is twofold:

- evaluation of the time for purchasing order emission;
- evaluation of the quantity to be ordered.

The inventory control policies implemented in the simulation model within each store and each distribution center are:

- the reorder point-order quantity policy;
- the reorder time-order quantity policy;
- the (s, S) policy;

Before starting to describe each policy, it is necessary to define notation to which authors will refer:

- $s_i(t)$, the re-order level at time t for the item i;
- $S_i(t)$, the target level at time t for the item i;
- $SS_i(t)$, the safety stock level at time t for the item i;
- $DF_i(t)$, the demand forecast at time t for the item i;
- $OHI_i(t)$, the on-hand inventory at time t for the item i;
- $OQ_i(t)$, the quantity already on order at time t for the item i;
- $SQ_i(t)$, the quantity to be shipped at time t for the item i;
- $Q_i(t)$, the quantity to be ordered at time t for the item i;
- $L_i(t)$, the lead time of the item i;
- $DFL_i(t)$, the demand forecast over the lead time for the item i;
- $IP_i(t)$, the inventory position at time t for the item i;

The inventory position $IP_i(t)$ is the on-hand inventory plus the quantity already on order minus the quantity to be shipped. In particular, it is defined as:

$$IP_i(t) = OH_i(t) + OQ_i(t) - SQ_i(t) \tag{1}$$

4.1. The mathematical model

In this section authors derive the mathematical model for each policy presented making reference to a single product *i*.

4.1.1. The reorder point-order quantity policy (RPOQ)

In this control policy the inventory level is continuously checked according to production/demand requirements. According to this policy, if $IP_i(t)$ falls below the $s_i(t)$, the purchase order has to be emitted. The quantity to be ordered can be defined using the Economic Order Quantity (*EOQ)* approach.

$$s_i(t) = DFL_i(t) + SS_i(t) \tag{2}$$

$$Q_i(t) = EOQ_i(t) \tag{3}$$

Such policy should be adopted when inventory level at each SC node is automatically monitored; there are no advantages in using scale economies; purchase orders can be regularly emitted.

4.1.2. The reorder time-order quantity policy (RTOQ)

Unlike the previous control policy, the reorder time-order quantity policy is based on a periodic check. If $T_i(t)$ is the review period of the item *i*, the quantity to order is defined by $S_i(t)$ minus $IP_i(t)$.

The value of $T_i(t)$ can be defined using the inverse formula usually used for evaluating the EOQ. In this policy, $S_i(t)$ represents the target level. This policy should be used when:

- inventory level is not automatically monitored;
- there are advantages related to scale economy;
- orders are not regular.

4.1.3. The ($s_i(t)$,$S_i(t)$) policy

This policy can be derived from the previous policies above mentioned. According to literature, there are two parameters which characterize this policy:

- $s_i(t)$, the re-order level at time t for the item i;
- $S_i(t)$, the target level at time t for the item i.

Authors introduce a new parameter, $K_i(t)$, a constant parameter which represents the average demand of the item *i* over a certain period of time.

The equations expressing $s_i(t)$ and $S_i(t)$ are as follows.

$$s_i(t) = DFL_i(t) + SS_i(t) \tag{4}$$

$$S_i(t) = s_i(t) + K_i(t) \tag{5}$$

Equations 6 and 7 respectively express the re-order condition and the quantity to be ordered.

$$IP_i(t) < s_i(t) \tag{6}$$

$$Q_i(t) = S_i(t) - IP_i(t) = s_i(t) + K_i(t) - IP_i(t) \tag{7}$$

5. INVENTORY POLICIES COMPARISON

Consider for each supply chain node the inventory control policies before described, different values of lead times, different level of demand intensity and demand variability as summarized in Table 1. The values of the lead times, demand intensity and demand variability are expressed as percentage of the actual values.

Table 1: Factors and Levels

Factors	L1	L2	L3
Lead Time	90%	100%	110%
Demand Intensity	90%	100%	110%
Demand Variability	90%	100%	110%

Simulation results, for each factors levels combination, are expressed in terms of average fill rate and on-hand inventory (i.e. the fill rate at store, for each item, is evaluated as ratio between the fully satisfied orders and the total quantity of orders). Simulation results are available for each store and for each distribution center. Let us consider the simulation results regarding one of the distribution center, similar results have been obtained for the remaining distribution center and stores. The following scenarios have been analyzed: (i) comparison of the 90%, 100% and 110% scenarios in terms of demand intensity; (ii) comparison of the 90%, 100% and 110% scenarios in terms of demand variability; (iii) comparison of the 90%, 100% and 110% scenarios in terms of lead times. For each scenario we investigated the behavior of the inventory control policies implemented in the simulation model.

Table 2 reports the simulation results in terms of fill rate for the first scenario.

Table 2: Simulation results comparison of the inventory control policies under different demand intensity – fill rate

Scenarios	RPOQ	RTOQ	sS
90% Demand Intensity	0.831	0.641	0.890
100% Demand Intensity	0.499	0.210	0.539
110% Demand Intensity	0.282	0.058	0.295

The best results in terms of fill rate are provided by the $s_i(t)$,$S_i(t)$ inventory control policy. The lowest value by the *RTOQ* inventory control policy. Note that the higher is the demand intensity the lower is the fill rate (for each inventory control policy). In addition for high demand intensity the *RPOQ* and $s_i(t)$,$S_i(t)$ inventory control policy show similar behaviors in terms of fill rate values.

Table 3 reports the simulation results in terms of on-hand inventory for the first scenario.

Table 3: Simulation results comparison of the inventory control policies under different demand intensity – on hand inventory

Scenarios	RPOQ	RTOQ	sS
90% Demand Intensity	100	113	54
100% Demand Intensity	105	121	69
110% Demand Intensity	154	194	134

Concerning the on-hand inventory the $s_i(t)$,$S_i(t)$ performs better than the other policies.

Table 4 reports the simulation results in terms of fill rate for the second scenario.

Table 4: Simulation results: comparison of the inventory control policies under different demand variability – fill rate

Scenarios	RPOQ	RTOQ	sS
90% Demand Variability	0.511	0.219	0.569
100% Demand Variability	0.496	0.205	0.533
110% Demand Variability	0.487	0.190	0.520

The *RTOQ* inventory control policy gives the worst performance. Note the similar behavior of the *RPOQ* and $s_i(t),S_i(t)$ policies. The policy based on the review period shows a better behavior in correspondence of low demand variability. In effect the higher is the demand variability the higher is the demand forecast error. The best policy with low demand variability is the $s_i(t),S_i(t)$, with the actual demand variability *RPOQ* and $s_i(t),S_i(t)$ show similar behavior, finally with high demand variability $s_i(t),S_i(t)$ allows to obtain the highest fill rate values.

Table 5 reports the simulation results in terms of on-hand inventory for the second scenario.

Table 5: Simulation results: comparison of the inventory control policies under different demand variability – on hand inventory

Scenarios	RPOQ	RTOQ	sS
90% Demand Variability	101	112	67
100% Demand Variability	104	113	69
110% Demand Variability	111	117	73

Concerning the on-hand inventory, once again, the $s_i(t),S_i(t)$ performs better than the other policies.

Table 6 reports the simulation results in terms of fill rate for the third scenario.

Table 6: Simulation results comparison of the inventory control policies under different lead time values – fill rate

Scenarios	RPOQ	RTOQ	sS
90% Lead Time	0.510	0.247	0.544
100% Lead Time	0.496	0.205	0.533
110% Lead Time	0.447	0.152	0.526

Such scenario investigates the effect of different lead times on the fill rate. Note that the higher is the lead time the lower is the fill rate (for each inventory control policy). The fill rate reduction passing from 90% to 100% lead time and from 100% to 110% are as follows: (i) 1.1% and 3.9% for the RPOQ control policy; (ii) 3.3% and 4.2% for the RTOQ control policy; (iii) 0.9% and 0.5% for the sS control policy. Consequently the $s_i(t),S_i(t)$ policy performs better than *RPOQ* and *RTOQ*.

Table 7 reports the simulation results in terms of on-hand inventory for the third scenario.

Table 7: Simulation results comparison of the inventory control policies under different lead time values – on hand inventory

Scenarios	RPOQ	RTOQ	sS
90% Lead Time	84	92	50
100% Lead Time	104	113	69
110% Lead Time	121	121	78

The simulation results analyzed in this section regard one of the distribution center. The variation of the demand intensity, of the demand variability and of the lead time allows to compare the different behaviors of the inventory control policy in order to find out the best policy in each situation both in terms of fill rate and in terms of on-hand inventory. Similar results have been obtained for each supply chain node, both retailers and distribution centers, analyzing the inventory systems along the supply chain.

6. CONCLUSIONS

The authors implemented a simulation model of a three echelons supply chain for studying the inventory management problem along the supply chain. Three different inventory control policies have been implemented in each node of the supply chain (on each store and distribution center excluding the manufacturing plant that work as make to order system without warehouse). The inventory control policies have been compared under different conditions in terms of demand intensity, demand variability and lead times, observing the variation of the fill rate and of the on-hand inventory. In particular three different values have been considered for the demand intensity, three for the demand variability and three for the lead time (in each case the middle value is the actual value). The simulation results analysis shows that the variation of the factors considered strongly affect the behavior of the inventory control policies, for each scenario the most suitable inventory control policy should be used. Further researches are still on going applying the genetic algorithms for evaluating optimal values for the parameters of both the demand forecast models and of the inventory control policies.

REFERENCES

Al-Rifai, M.H., Rossetti, M.D., 2007. An efficient heuristic optimization algorithm for a two-echelon (R, Q) inventory system. *Production Economics*, 109, 195–213.

Bhaskaran, S., 1998. Simulation analysis of a manufacturing supply chain. *Decision Sciences*, 29 (3), 633–657.

Bruzzone A.G., Mosca R., Briano C., Brandolini M. 2000. Models for Supporting Customer Satisfaction in Retail Chains. *Proceedings of HMS, Portofino, October 5-7.*

Bruzzone A.G., Massei M., Brandolini M., 2006. Simulation Based Analysis on Different Logistics Solutions for Fresh Food Supply Chain. *Proceedings of SCSC2006, Calgary, Canada.*

Bruzzone A.G., Bocca E. 2006. Logistics and Process Solutions for Supply Chain of Fresh Food In Retail. *Proceedings of HMS2006, Barcelona.*

Bruzzone, A.G., Longo, F., Brandolini, M., 2006. Enhancement Process Based on Simulation Supporting Efficiency & Organization, *Proceedings of Summer Computer Simulation*

Conference, July 30th – August 03rd, Calgary Canada.

Bruzzone, A.G., Longo, F., Viazzo, S., Mirabelli, G., Papoff, E., Massei, M., Briano, C., 2004. Discrete event simulation applied to modelling and analysis of a supply chain. *Proceedings of Modeling and Applied Simulation Conference*, Bergeggi (SV) (Italy).

Chen, C., Lee, W., 2004. Multi-objective optimization of multi-echelon supply chain networks with uncertain product demands and prices. *Computers and Chemical Engineering*, 28,1131–1144.

Chen, F., Krass, D., 2001. Inventory models with minimal service level constraints. *Operational Research*, 134, 120–140.

Curcio, D., Longo, F., Mirabelli, G., 2007. Inventory policies comparison in a manufacturing process using modeling & simulation. *Proceedings of the International Mediterranean Modelling Multiconference,* 237 – 242. October 04 – 06, Bergeggi (Italy).

Dellaert, N., De Kok, T., 2004. Integrating resource and production decisions in a simple multi-stage assembly system. *Production Economics*, 90, 281–294.

De Sensi, G. , Longo, F., Mirabelli, G., 2007. Inventory policies analysis under demand patterns and lead times constraints in a real supply chain. *Production Research*.

D'Esopo, A., 1968. An ordering policy for stock items when delivery can be expedited. *Operations Research*, 16 (4), 880–883.

Giannoccaro, I., Pontrandolfo, P., 2002. Inventory management in supply chains: a reinforcement learning approach. *Production Economics*, 78, 153–161.

Hau, L.L., Billington, C., 1993. Material Management in Decentralized Supply Chains. *Operations Research*, 41(5), 835–847.

Hsu, V., Lee,C., So, K., 2006. Optimal Component Stocking Policy for Assemble-to-Order Systems with Lead-Time-Dependent Component and Product Pricing. Management Science, 52 (3), 337–351.

Huang, H.C., Chew, E.P., Goh, K.H., 2005. A two-echelon inventory system with transportation capacity constraint. *Operational Research*, 167, 129–143.

Inderfurth, K., Minner, S., 1998. Safety stocks in multi-stage inventory systems under different service measures. *Operational Research*, 106, 57–73.

Lee, H.T., Wu, J.C., 2006. A study on inventory replenishment policies in a two-echelon supply chain system. *Computers & Industrial Engineering*, 51, 257–263.

Longo, F., Mirabelli, G., Papoff, E., 2006. Material Flow Analysis and Plant Lay-Out Optimization of a Manufacturing System. *International Journal of Computing*, 5(1), 107–116.

Longo, F., Mirabelli, G., 2008. An Advanced Supply Chain Management Tool Based on Modeling & Simulation, *Computer and Industrial Engineering*, 54 (3), 570–588.

Longo, F., Mirabelli, G., Papoff, E. Modeling Analysis and Simulation of a supply chain devoted to support pharmaceutical business retail, *Proceedings of the 18th International Conference on Production Research, July 31 – August 4 , Salerno, Italy.*

Minner, S., 2003. Multiple-supplier inventory models in supply chain management: A review. *Production Economics*, 81, 265–279.

Moinzadeh, K., 2001. An improved ordering policy for continuous review inventory systems with arbitrary inter-demand time distributions, *IIE Transactions*, 33, 111–118.

Qi, X., Bard, J., Yu, G., 2004. Supply chain coordination with demand disruptions. *Omega*, 32, 301–312.

Ramasesh, R., Keith, O., Hayya, J., 1991. Sole versus Dual Sourcing in Stochastic Lead-Time (s, Q) Inventory Model. *Management Science*, 37 (4), 428–443.

Zhou, B., Zhao, Y., Katehakis, M., 2007. Effective control policies for stochastic inventory systems with a minimum order quantity and linear costs. *Production Economics*, 106, 523–531.

AUTHORS BIOGRAPHIES

DUILIO CURCIO was born in Vibo Valentia (Italy), on December the 15th, 1981. He took the degree in Mechanical Engineering from University of Calabria (2006). He is currently PhD student at the Mechanical Department of University of Calabria. His research activities include Modeling & Simulation and Inventory Management theory for production systems and Supply Chain design and management. He collaborates with the Industrial Engineering Section of the University of Calabria to research projects for supporting Research and Development in SMEs.

FRANCESCO LONGO was born in Crotone (Italy), on February the 08th, 1979. He took the degree in Mechanical Engineering from University of Calabria (2002). He received the PhD in Industrial Engineering (2005). He is currently researcher at the Mechanical Department (Industrial Engineering Section) of University of Calabria and scientific responsible of the Modeling & Simulation Center – Laboratory of Enterprise Solutions (MSC-LES) in the same department. His research interests regard modeling & simulation of manufacturing systems and supply chain, DOE, ANOVA.

AN APPROACH FOR FAULT DETECTION IN DEVS MODELS

Diego M. Llarrull[(a)], **Norbert Giambiasi** [(b)]

[(a)] CIFASIS - CCT-CONICET. 27 de Febrero 210 bis. S2000EZP - Rosario
[(b)] LSIS - UMR CNRS 6168. University Paul Cézanne

[(a)] diego.llarrull@lsis.org, [(b)] norbert.giambiasi@lsis.org

ABSTRACT

The aim of this paper is to present a first approach for building distinguishing, homing, and synchronizing sequences for a subset of DEVS models, in order to apply to them the fault detection techniques developed on Mealy machines. After the definition of the considered subset of DEVS (called MealyDEVS), we present the extension of fault detection techniques on this DEVS subset.

Keywords: DEVS, fault detection, Mealy machines, black-box testing.

1. INTRODUCTION

The design of real-time discrete event control systems is a process that requires dedicated formalisms and adapted tools. In particular, the DEVS formalism is convenient for the low-level phase of the design process since it provides a suitable simulation framework, thus enabling the possibility of validating models by simulation.

In (Dacharry and Giambiasi 2005), a formal methodology for the design and verification of control systems is presented. With this methodology, a high-level specification of a system to be designed is constructed using a network of timed automata, and the corresponding implementation is expressed as a coupled DEVS model. This makes it possible to formally verify the conformance of critical components (atomic DEVS models against timed automata) and the conformance of the whole model. Nevertheless, due to the state explosion problem that frequently appears in the verification of models that deal with a dense time base, the automatic verification of the conformance between the high-level and the low-level models is unfeasible in the general case.

Despite this discouraging result, a partial automatic validation of the conformance relation between an implementation and its specification is possible, for example, by generating test cases on a high-level specification and applying these tests to the low-level model description. Several formalisms have a developed theory of fault detection techniques. Most of them are related to Mealy Machines, which have been widely used for testing purposes in various domains. Mealy Machines are based on the hypothesis of simultaneous input/output events and are untimed models.

Our proposal is then to allow the use of fault detection techniques on timed models of higher complexity. In this paper we will extend the theory presented by (Kohavi 1978) to a subset of the models that can be represented using the DEVS formalism.
The paper is organized as follows: in Section 1 we recall the existing theory, together with the tools and concepts that will be necessary to extend it. In Section 2, we introduce a subset of the DEVS formalism that we take under consideration, and we adapt and extend the existing methods, concepts and definitions to this subset. In Section 3, we propose an extension of the first subset of models in order to enlarge the spectrum of models to which the theory of fault detection can be applied, and we briefly show some considerations about the implementation of these testing methods. Finally, we conclude the paper.

2. PRELIMINARIES

2.1. Mealy Machines

A Mealy Machine (Kohavi 1978, Lee et al 1996) is formally stated as a quintuple $M=(I,O,S,\delta,\lambda)$ where *I*, *O* and *S* are finite and nonempty sets of input symbols, output symbols, and states respectively, $\delta : S\times I \to S$ is the state transition function and $\lambda : S\times I \to O$ is the output function.

When the machine is in a current state $s_i \in S$ and receives an input $a\in I$ it moves to the next state s_j specified by $\delta(s_i,a)=s_j$ and produces immediately the output y specified by $\lambda(s_i,a)=y$.

2.1.1. Execution fragments and Traces

In the following, some concepts that will be useful in the subsequent sections are recalled (Kohavi 1978, Lynch and Vaandrager 1993a, Lynch and Vaandrager 1993b) in order to adjust them to the syntax used by the DEVS formalism and the concepts of executions and traces as defined in (Dacharry and Giambiasi 2005).

Definition 2.1 (Execution fragment) *Let* $\mathcal{M}=(X_{\mathcal{M}},Y_{\mathcal{M}},S_{\mathcal{M}},\delta_{\mathcal{M}},\lambda_{\mathcal{M}})$ *be a Mealy machine. Then an execution fragment for M is a finite or infinite alternating sequence of the form* $s_0,x_0,y_0,s_1,x_1,y_1,...$ *beginning with a state (and if it is finite also ending with a state), such that* $\forall i\in\{0..n\}\bullet x_i \in X_{\mathcal{M}}, y_i \in Y_{\mathcal{M}}, s_i \in S_{\mathcal{M}},\ \delta_{\mathcal{M}}(s_i,x_i)=s_{i+1} \ \wedge$ $\lambda_{\mathcal{M}}(s_i,x_i)=y_i$

Definition 2.2 (Execution of a Mealy machine) *Let* $\mathcal{M}=(X_{\mathcal{M}},Y_{\mathcal{M}},S_{\mathcal{M}},\delta_{\mathcal{M}},\lambda_{\mathcal{M}})$ *be a Mealy machine. Then an execution for* $\mathcal{M}$ *is a finite execution fragment of* $\mathcal{M}$ *that begins with a starting state. We denote with* $execs^{*}(\mathcal{M})$, $execs^{\omega}(\mathcal{M})$, *and execs(M) the sets of finite, infinite, and all executions of* $\mathcal{M}$, *respectively*

Definition 2.3 (Reachability of a state) *Let* $\mathcal{M}=(X_{\mathcal{M}},Y_{\mathcal{M}},S_{\mathcal{M}},\delta_{\mathcal{M}},\lambda_{\mathcal{M}})$ *be a Mealy machine, and* $\textbf{last}$ *a function such that* $\textbf{last}(p)$ *is the last element of the finite sequence* π. *Then a state* $s_i \in S_{\mathcal{M}}$ *is reachable if* $s=\textbf{last}(\alpha)$ *for some finite execution* α *of* $\mathcal{M}$

Definition 2.4 (Trace) *Let* $\mathcal{M}=(X_{\mathcal{M}},Y_{\mathcal{M}},S_{\mathcal{M}},\delta_{\mathcal{M}},\lambda_{\mathcal{M}})$ *be a Mealy machine,* α *an execution fragment of* $\mathcal{M}$ *and* $\textbf{states}(\alpha)$ *the set of states that appear in* α. *Let* γ *be the sequence consisting of the events in* α. *Then* $\textbf{trace}(\alpha)$ *is defined to be the tuple* (θ_I,θ_O) *consisting of the members of* γ, *where* $\forall a_{O_j} \in \theta_O, a_{I_j} \in \theta_I, s_j \in states(\alpha) \bullet \lambda_{\mathcal{M}}(s_j, a_{I_j}) = a_{O_j}$ *Moreover, a finite or infinite tuple* β *is a trace of* $\mathcal{M}$ *if* $\mathcal{M}$ *has an execution* α *with* $\beta = traces(\alpha)$. *We denote with* $\textbf{traces}^{*}(\mathcal{M})$, $\textbf{traces}^{\omega}(\mathcal{M})$ *and* $\textbf{traces}(\mathcal{M})$ *the sets of finite, infinite, and all traces of* $\mathcal{M}$, *respectively.*

2.1.2. **Minimality - State Equivalence** (Kohavi 1978)

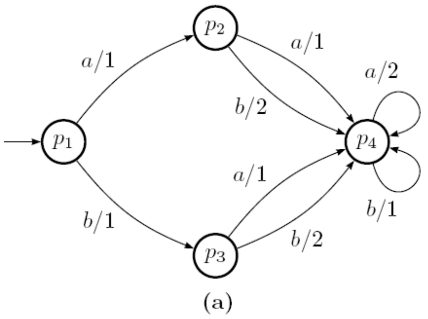
Definition 2.5 (Distinguishing sequence) *Two states* s_i *and* s_j *of a Mealy machine* $\mathcal{M}$ *are distinguishable if and only if there exist at least two execution fragments* α *and* β *of* $\mathcal{M}$ *with* $\textbf{traces}(\alpha)=(\theta_{I_\alpha},\theta_{O_\alpha})$, $\textbf{traces}(\beta)=(\theta_{I_\beta},\theta_{O_\beta})$ *where* $first(\alpha) = s_i$, $first(\beta)=s_j$, $\theta_{I_\alpha}=\theta_{I_\beta}$ *and* $\theta_{O_\alpha}\neq\theta_{O_\beta}$. θ_{I_α} $(=\theta_{I_\beta})$ *is called a distinguishing sequence of the pair* (s_i,s_j). *If there exists for pair* (s_i,s_j) *a distinguishing sequence of length k, then the states in* (s_i,s_j) *are said to be k-distinguishable.*

Definition 2.6 (k-equivalence of states) *Two states* s_i *and* s_j *of a Mealy machine M are k-equivalent if and only if they are not k-distinguishable.*

Definition 2.7 (Equivalence of states) *Two states* s_i *and* s_j *of a Mealy machine* $\mathcal{M}$ *are equivalent if and only if they are k-equivalent* $\forall k \in \Re_0^{+}$. *In other words,* s_i *and* s_j *are equivalent if and only if, for every possible input sequence, the same output sequence will be produced regardless of whether* s_i *or* s_j *is the initial state.*

Definition 2.8 (Equivalence of Mealy machines) *Two Mealy machines* $\mathcal{M}_1$ *and* $\mathcal{M}_2$ *are equivalent if and only if, for every state in* $\mathcal{M}_1$, *there is a corresponding equivalent state in* $\mathcal{M}_2$, *and vice-versa.*

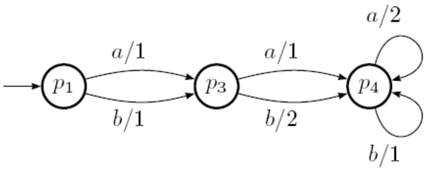
Definition 2.9 (Minimal Mealy machine) *A Mealy Machine is minimal (reduced) if and only if no two states in it are equivalent. Additionally, the Mealy machine* $\mathcal{M}_1$ *which contains no equivalent states and is equivalent to the Mealy machine* $\mathcal{M}_2$ *is said to be the minimal, or reduced form of* $\mathcal{M}_2$.



(a)

State	$x=a$	$x=b$
p_1	$p_2, 1$	$p_3, 1$
p_2	$p_4, 1$	$p_4, 2$
p_3	$p_4, 1$	$p_4, 2$
p_4	$p_4, 2$	$p_4, 1$

(b)



(c)

Phase	$x=a$	$x=b$
p_1	$p_3, 1$	$p_3, 1$
p_3	$p_4, 1$	$p_4, 2$
p_4	$p_4, 2$	$p_4, 1$

(d)

Figure 1: Mealy machine (a) with its associated transition table (b). Resulting minimal Mealy machine (c) after the deletion of p_2 which is equivalent to p_3 and the resulting transition table (d).

2.2. Labelled Timed Transition Systems

Definition 2.10 (Labelled Timed Transition System) A labelled timed transition system $\mathcal{T}_t$ *is an automaton whose alphabet includes* $\Re^{+}$. *The transitions* *corresponding to symbols from $\Re^+$ are referred to as time-passage transitions, while non-time-passage transitions are referred to as discrete transitions. So, a labelled timed transition system consists of:*

S *a possibly infinite set of states,*

INIT *an initial state,*

Σ *a set of discrete actions,*

D *a set of discrete transitions, noted* $s \xrightarrow{x} s'$*, where* $x \in \Sigma_{\mathcal{T}_t}$ *and* $s, s' \in S_{\mathcal{T}_t}$*, asserting that*

> *"from state s the system can instantaneously move to state s' via the occurrence of the event x", and*

T *a set of time-passage transitions, noted* $s \xrightarrow{t} s'$*, where* $t \in \Re^+$ *and* $s, s' \in S_{\mathcal{T}_t}$*, asserting that*

> *"from state s the system can move to state s' during a positive amount of time t in which no discrete events occur".*

A labelled timed transition system is assumed to satisfy two axioms.

S1 *If* $s \xrightarrow{t} s'$ *and* $s' \xrightarrow{t'} s''$*, with* $t, t' \in \Re^+$*, then* $s \xrightarrow{t+t'} s''$.

S2 *Each time-passage step,* $s \xrightarrow{t} s'$*, with* $t \in \Re^+$*, has a trajectory, where a trajectory describes the state changes than can occur during time-passage transitions. If I is any closed interval of* $\Re_0^+$ *beginning with 0, an I–trajectory is defined as a function,* $v : I \to S$ *such that:*

$$v(t) \xrightarrow{t-t'} v(t') \;\; \forall t, t' \in I \,|\, t < t'$$

It will be useful for our purposes to abstract away the quantitative aspect of time in LTSs. The relationship concerned is called *Delay Time-Abstracting Bisimulation* (Tripakis 2001).

Definition 2.11 (Delay Time-Abstracting Bisimulation) *Consider a labelled timed transition system A with sets of discrete transitions D and time-passage transitions E. A binary relation ≈ on the states of A is a delay time-abstracting bisimulation (DTaB) if, for all states* $s_1 \approx s_2$*, the following conditions hold:*

1. if $s_1 \xrightarrow{d_1} s_3$, for some $d_1 \in D$ then there exists $\delta_1 \in E$ and $d_2 \in D$ such that $s_2 \xrightarrow{\delta_1} \xrightarrow{d_2} s_4$ and $s_3 \approx s_4$;
2. if $s_1 \xrightarrow{\delta_1} s_3$, for some $\delta_1 \in E$ then there exists $\delta_2 \in E$ such that $s_2 \xrightarrow{\delta_2} s_4$ and $s_3 \approx s_4$;
3. The above conditions also hold if the roles of s_1 and s_2 are reversed.

Then, the states s_1 *and* s_2 *are said to be DTa-bisimilar. In general, two TTSs* $\mathcal{A}_1$ *and* $\mathcal{A}_2$ *are said to be DTa-bisimilar if there exists a DTaB ≈ on the states of* $\mathcal{A}_1$ *and* $\mathcal{A}_2$ *such that* $s_0^1 \approx s_0^2$*, where* s_0^i *is the initial state of* $\mathcal{A}_i$.

2.3. DEVS formalism

A DEVS model (Zeigler 2000) is a structure $M = \langle X, S, Y, \delta_{int}, \delta_{ext}, \lambda, ta \rangle$ where

- *X* is the set of input values
- *S* is a set of states,
- *Y* is the set of output values
- $\delta_{int} : S \to S$ is the internal transition function
- $\delta_{ext} : Q \times X \to S$ is the external transition function, where
- $Q = \{(s,e) \,|\, s \in S, 0 \le e \le ta(s)\}$ is the total state set
- *e* is the time elapsed since the last transition
- $\lambda : S \to Y$ is the output function
- $ta : S \to \Re_0^+ \cup \infty$ is the time advance function.

The interpretation of these elements is the following: at any time the model is in some state, *s*. If no external event occurs the model stay in state *s* for time *ta*(*s*). Notice that *ta*(*s*) could be a real number. But it can also take on the values 0 and ∞. Depending on the value of *ta*(*s*), an atomic DEVS model has two kinds of states:

- *Passive States:* A state $s \in S$ is called *passive* iff *ta*(*s*) = ∞ and no internal transition is defined in it. The set of all passive states of a DEVS model is referred to as S_p.
- *Active States:* A state $s \in S$ is called *active* iff *ta*(*s*) ≠ ∞. The set of all active states of a DEVS model is referred to as S_a.

When the elapsed time in the current state, *e,* equals *ta*(*s*), the system outputs the value, λ(*s*), and changes to state $\delta_{int}(s)$. If an external event *x* occurs before *e = ta(s)*, the model transits into the state $\delta_{ext}(s, e, x)$.

The dense-time characteristics of DEVS models impose a restriction in the tractability of the problem of fault detection techniques in the most general case. It is then necessary to reduce the subset of DEVS models to those models where these techniques can be applied in an efficient way.

2.3.1. Execution fragments and Traces for DEVS model (Giambiasi and Dacharry 2007):

Definition 2.12 (Execution fragment of a DEVS model) *Let* $\mathcal{D} = (X_{\mathcal{D}}, Y_{\mathcal{D}}, S_{\mathcal{D}}, \delta_{ext\mathcal{D}}, \delta_{int\mathcal{D}}, \lambda_{\mathcal{D}}, ta_{\mathcal{D}})$ *be a DEVS model. Then an execution fragment for D is a finite alternating sequence* $\Upsilon = v_0\, x_1\, v_1\, x_2\, v_2 \ldots x_n\, v_n$ *where:*

- **time-passage transitions:** Each v_i is a function from a real interval $I_i = [0, t_i]$ to the set of total phases of *D*, such that $\forall j, j' \in I_i \mid j < j'$, if $v_i(j) = (s, e)$ then $v_i(j') = (s, e + j' - j)$
- **event transitions:** Each x_i is an input or output event, and if $(s, e) = v_{i-1}(sup(I_{i-1}),$ $(s', 0) = v_i(inf(I_i))$, one of the following conditions hold:

1. $x_i \in Y_{\mathcal{D}}, \delta_{int\mathcal{D}}(s) = s'$, ta(s)=e, and $\lambda(s) = x_i$.
2. $x_i \in X_{\mathcal{D}}, \delta_{ext\mathcal{D}}(s, e, x_i) = s'$, and e ≤ ta(s).

The definitions of execution and state reachability of DEVS models are analogous to those of Mealy machines. However, due to the timed nature of DEVS models, the definitions of traces and distinguishability of states are more complex (Giambiasi and Dacharry 2007):

Definition 2.13 (Trace of a DEVS) *Let* $\mathcal{D} = (X_{\mathcal{D}}, Y_{\mathcal{D}}, S_{\mathcal{D}}, \delta_{ext\mathcal{D}}, \delta_{int\mathcal{D}}, \lambda_{\mathcal{D}}, ta_{\mathcal{D}})$ *be a DEVS model,* $\Upsilon = v_0\, x_1\, v_1\, x_2\, v_2 \ldots x_n\, v_n$ *an execution fragment of D. Then trace*(Υ) *is defined to be a tuple* (θ_I, θ_O, t) *such that* θ_I *and* θ_O *are sequences consisting of all pairs of events of* Υ *and their time of occurrence, sorted in chronological order of occurrence, and t is the total time of execution, defined as* $\sum_{0 \le j \le n} (sup(I_j))$*. Formally, the time of occurrence of an event* x_i *of* Υ *is equal to* $\sum_{0 \le j < i} (sup(I_j))$*, with* I_j *the domain of* v_j*.*

The set of all traces of a DEVS model is defined as $traces(\mathcal{D}) = \{\, trace(\Upsilon) \mid \Upsilon \in execs(\mathcal{D}) \,\}$.

2.3.2. Associated Transition System for a DEVS model

The semantics of a DEVS model can be clearly stated by means of its *associated Timed-Transition System* (Giambiasi and Dacharry 2007).

Definition 2.14 (Associated Transition System) *Given a DEVS model D its associated transition system is defined over the alphabet* $\Sigma = X_{\mathcal{D}} \cup Y_{\mathcal{D}}$ *, Taut(D) as a labeled timed transition system* $\mathcal{T}_t$*, where:*

1. *the set of states,* $S_{\mathcal{T}_t}$*, consists of the set of total phases of D ,* $Q_{\mathcal{D}}$*,*
2. *the initial phase, init(*$\mathcal{T}_t$*) is (s,0), where* $s \in S_{\mathcal{D}}$*, and s is the discrete phase defined as the initial phase of the DEVS model,*
3. *the set of discrete transitions,* $D_{\mathcal{T}_t}$

   $D_{\mathcal{T}_t} = \{(s, e) \xrightarrow{x} (s', 0) \mid (\delta_{int}(s) = s'$
   $\wedge \lambda(s) = x \wedge e = ta(s))$ *or*
   $(\delta_{ext}((s, e), x) = s' \wedge e \le ta(s))\}$
4. *the set of time-passage transitions,* $T_{\mathcal{T}_t}$ *,*
5. $T_{\mathcal{T}_t} = \{(s, e) \xrightarrow{t} (s, e') \mid (s, e) \in Q_{\mathcal{D}}, e' = e + t,$
   $0 \le e + t \le ta(s)\}$

The labelled timed transition system associated with a DEVS model, as defined above, specifies the same set of traces as its corresponding DEVS model (Giambiasi and Dacharry 2007).

3. MEALYDEVS

The basic idea for constructing a DEVS model which behaves exactly like a Mealy machine is that of forcing an immediate input/output response. As a consequence, the subset of DEVS models that we consider first consists only of such models where $\forall s \in S_a \Rightarrow ta(s)=0$. For the models of this subset, all active states are *transitory*. Transitory states are said to be *input-blocking* (Giambiasi and Dacharry 2005). That is, the MealyDEVS model does not stay in an active state; it appears (externally) to be always ready for input. Then, every state transition of a Mealy machine $\mathcal{M} = (I_{\mathcal{M}}, O_{\mathcal{M}}, S_{\mathcal{M}}, \delta_{\mathcal{M}}, \lambda_{\mathcal{M}})$ that has the form

$s_i \xrightarrow{x/y} s_j$, where $x \in I_{\mathcal{M}}$, $y \in O_{\mathcal{M}}$, $s_i, s_j \in S_{\mathcal{M}}$

and *x*/*y* means that the input event *x* is received and the output event *y* is sent in this transition, is translated into two transitions in the corresponding DEVS model $\mathcal{D} =< X_{\mathcal{D}}, Y_{\mathcal{D}}, S_{\mathcal{D}}, \delta_{int_{\mathcal{D}}}, \delta_{ext_{\mathcal{D}}}, \lambda_{\mathcal{D}}, ta_{\mathcal{D}} >$:

- An external transition of the form $s_i \xrightarrow{x/-} s_{i,x}$ where $\delta_{ext_{\mathcal{D}}}(s_i, e, x) = s_{i,x}$, $s_i \in S_{s_{\mathcal{D}}}$, $e \in \Re_0^+ \cup \{\infty\}$, $x \in X_{\mathcal{D}}$ and $s_{i,x} \in S_{a_{\mathcal{D}}}$.
- An internal transition of the form $s_{i,x} \xrightarrow{-/y} s_j$ where $\delta_{int_{\mathcal{D}}}(s_{i,k}) = s_j$, $\lambda(s_{i,k}) = y$, $s_{i,k} \in S_{a_{\mathcal{D}}}$, $y \in Y_{\mathcal{D}}$ and $s_j \in S_{s_{\mathcal{D}}}$.

This translation is possible provided the following constraints are satisfied: firstly, it is needed that both models have the same input and output event sets (that is, $I_{\mathcal{M}} = X_{\mathcal{D}} \wedge O_{\mathcal{M}} = Y_{\mathcal{D}}$), and the set of passive states of *D* has to be equal to the set of all states of *M* ($S_{\mathcal{M}} = S_{s_{\mathcal{D}}}$). It is also required that for each possible transition $\delta_{\mathcal{M}}(s_i, x) = s_j$ of *M* there exist two transitions in *D* , $\delta_{ext_{\mathcal{D}}}(s_i, e, x)$ and $\delta_{int_{\mathcal{D}}}(s_{i,x})$ such that $\delta_{\mathcal{M}}(s_i, x) = s_j \Leftrightarrow \forall e \in \Re_0^+ \cup \{\infty\} \bullet \delta_{ext_{\mathcal{D}}}(s_i, e, x) = s_{i,x}$ and $\delta_{int_{\mathcal{D}}}(s_{i,x}) = s_j$. Finally, for each possible output event $y \in I_{\mathcal{M}}$ such that $\lambda_{\mathcal{M}}(s_i, x) = y$ it is required an analogous output event in *D* (in symbols, $\lambda_{\mathcal{M}}(s_i, x) = y \Leftrightarrow \lambda_{\mathcal{D}}(s_{i,x}) = y$).

Note that in this subset of atomic DEVS models the external transition function *does not depend on the elapsed time*. That is, $\forall s_i \in S_{p\mathcal{D}}; e, e' \in \Re_0^+ \cup \{\infty\} \bullet$ $\delta_{ext\mathcal{D}}(s_i, e, x) = \delta_{ext\mathcal{D}}(s_i, e', x)$.

The subset of DEVS models which capture this behaviour is called MealyDEVS and is formally defined as:

Definition 3.1 (MealyDEVS) *A DEVS model* $\mathcal{D} = (X_{\mathcal{D}}, Y_{\mathcal{D}}, S_{\mathcal{D}}, \delta_{ext\mathcal{D}}, \delta_{int\mathcal{D}}, \lambda_{\mathcal{D}}, ta_{\mathcal{D}})$ *is a MealyDEVS model if and only if*

- $S_{\mathcal{D}} = S_{a\mathcal{D}}$ *(active states)* $\cup\, S_{p\mathcal{D}}$ *(passive states).*
- $\delta_{ext\mathcal{D}} : S_{p\mathcal{D}} \times \Re_0^+ \cup \{\infty\} \times X \to S_{a\mathcal{D}}$ *where*
- $\forall s_i \in S_{p\mathcal{D}}; e, e' \in \Re_0^+ \cup \{\infty\} \bullet$ $\delta_{ext\mathcal{D}}(s_i, e, x) = \delta_{ext\mathcal{D}}(s_i, e', x)$
- $\delta_{int\mathcal{D}} : S_{a\mathcal{D}} \to S_{p\mathcal{D}}$
- $\lambda_{\mathcal{D}} : S_{a\mathcal{D}} \to Y_{\mathcal{D}}$ *where*
  $\forall s_j \in S_{a\mathcal{D}} \bullet ta_{\mathcal{D}}(s_j) = 0$
  (All active states are transitory)

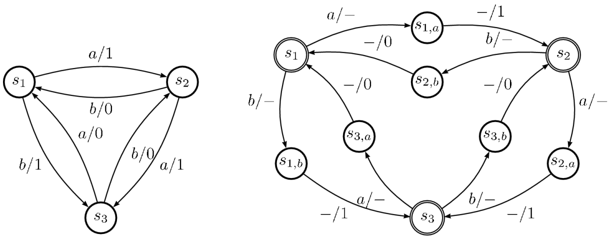


Figure 2: Transition diagram for a Mealy machine and its corresponding MealyDEVS model.

We consider only *completely specified* models.

It is straightforward that every Mealy machine has the same input/output behaviour as its corresponding MealyDEVS model.

Remark: The previous definitions and procedures can be applied to untimed DEVS (Giambiasi and Dacharry 2007).

4. EXTENDED MEALYDEVS

It should be clear that the MealyDEVS subset represents a tiny subset of the systems than can be modelled using the DEVS formalism. It is then of major interest to expand this subset in order to apply fault detection techniques for a wider range of DEVS models.

We consider now a subset called Extended MealyDEVS. In this subset, the considered DEVS models have *a time advance function which can take arbitrary finite values on active states* (in symbols: $\forall s_i \in S_{a\mathcal{D}} \bullet \ ta(s_i) \ \in \Re_0^+$).

Definition 4.1 (Extended MealyDEVS) *A DEVS model* $\mathcal{D} = (X_{\mathcal{D}}, Y_{\mathcal{D}}, S_{\mathcal{D}}, \delta_{ext\mathcal{D}}, \delta_{int\mathcal{D}}, \lambda_{\mathcal{D}}, ta_{\mathcal{D}})$ *is an Extended MealyDEVS if and only if*

- $S_{\mathcal{D}} = S_{a\mathcal{D}} \cup S_{p\mathcal{D}}$ where
  $S_{a\mathcal{D}} \cap S_{p\mathcal{D}} = \varnothing$
- $\delta_{ext\mathcal{D}} : S_{p\mathcal{D}} \times \Re_0^+ \cup \{\infty\} \times X \to S_{a\mathcal{D}}$ where
  $\forall s_i \in S_{p\mathcal{D}}; e, e' \in \Re_0^+ \cup \{\infty\} \bullet$
  $\delta_{ext\mathcal{D}}(s_i, e, x) = \delta_{ext\mathcal{D}}(s_i, e', x)$
- $\delta_{int\mathcal{D}} : S_{a\mathcal{D}} \to S_{p\mathcal{D}}$
- $\lambda_{\mathcal{D}} : S_{a\mathcal{D}} \to Y_{\mathcal{D}}$
- $ta_{\mathcal{D}} : S_{\mathcal{D}} \to \Re_0^+ \cup \{\infty\}$ where
  $\forall s_i \in S_{p\mathcal{D}} \bullet ta_{\mathcal{D}}(s_i) = \infty$ and
  $\exists k \in \Re_0^+ \mid \forall s_j \in S_{a\mathcal{D}} \bullet ta_{\mathcal{D}}(s_j) < k$ (ta takes finite real values on active states).

It is necessary to define a special kind of timed input sequence so that it ensures that the input events always occur when the model is in a passive state. In order to ensure that a model $\mathcal{D}$ is in a passive state, the input has to be *delayed* for at least t_k units of time, where t_k = min $\{x \in \Re^+ \mid > t\}$ and $t = max\{ta(s_i) \mid s_i \in S_{a\mathcal{D}}\}$, which is the maximum time that can elapse before the model reaches another passive state. Such a sequence is called a *slow timed input sequence*:

Definition 4.2 (Slow timed sequence) *Let* $\mathcal{D} =< X_{\mathcal{D}}, Y_{\mathcal{D}}, S_{\mathcal{D}}, \delta_{int_{\mathcal{D}}}, \delta_{ext_{\mathcal{D}}}, \lambda_{\mathcal{D}}, ta_{\mathcal{D}} >$ *be a DEVS model, where* $S_{a\mathcal{D}} \subset S_{\mathcal{D}}$. *A timed sequence for* $\mathcal{D}$ *is a finite series of the form* $<(x_0, t_0), (x_1, t_1), .., (x_n, t_n)>$ *such that*
$\forall i \in \{0..n\} \bullet x_i \in X_{\mathcal{D}} \wedge \forall i \in \{0..n\} \bullet t_i \in \Re_0^+ \wedge t_{i+1} - t_i > k$
where $k = max(ta_{\mathcal{D}}(s_{a_i}) \mid s_{a_i} \in S_{a\mathcal{D}})$

In order to formally explicit the relation between a MealyDEVS model and an Extended MealyDEVS model, we show subsequently the existence of a *Delay Time-Abstracting Bisimulation* (Tripakis and Yovine 2001) between them.

Theorem 4.1: *The MealyDEVS model* $\mathcal{D} = (X_{\mathcal{D}}, Y_{\mathcal{D}}, S_{\mathcal{D}}, \delta_{ext\mathcal{D}}, \delta_{int\mathcal{D}}, \lambda_{\mathcal{D}}, ta_{\mathcal{D}})$ *is DTa-bisimilar to the Extended MealyDEVS model* $\mathcal{E} = (X_{\mathcal{D}}, Y_{\mathcal{D}}, S_{\mathcal{D}}, \delta_{ext\mathcal{D}}, \delta_{int\mathcal{D}}, \lambda_{\mathcal{D}}, ta_{\mathcal{E}})$ *which only differs from D in the values of ta(s).*

Proof: Since $\mathcal{D}$ and $\mathcal{E}$ have identical $\delta_{ext\mathcal{D}}$ and $\delta_{int\mathcal{D}}$ functions, it is straightforward that:

- For each time-passage transition of the form $s^{\mathcal{D}} \xrightarrow{\delta_1} s^{\mathcal{D}}$ in $\mathcal{D}$, where $\delta_1 \in \Re_0^+$ there is a corresponding transition $s^{\mathcal{E}} \xrightarrow{\delta_1} s^{\mathcal{E}}$ in $\mathcal{E}$, where s is a passive state and since there are no internal transitions in this kind of states.
- For each discrete transition of the form $s_0^{\mathcal{D}} \xrightarrow{x_i} s_{0,x_i}^{\mathcal{D}}$ where $x_i \in X_{\mathcal{D}}\ \forall i = 1..n_x$ there is a corresponding discrete transition of the form $s_0^{\mathcal{E}} \xrightarrow{x_i} s_{0,x_i}^{\mathcal{E}}$ in $\mathcal{E}$.
- For each discrete transition of the form $s_{0,x_i}^{\mathcal{D}} \xrightarrow{y_i} s_{x_{i+1}}^{\mathcal{D}}$ where $y_i \in Y_{\mathcal{D}}\ \forall i = 1..n_y$ there is a pair composed of a time-passage transition of the form $s_{0,x_i}^{\mathcal{E}} \xrightarrow{\delta_2} s_{0,x_i}^{\mathcal{E}}$, where $\delta_2 = ta_{\mathcal{E}}(s_{0,x_i}^{\mathcal{E}})$ and the corresponding discrete transition of the form $s_{0,x_i}^{\mathcal{E}} \xrightarrow{y_i} s_{x_{i+1}}^{\mathcal{E}}$.

Then, the relation $\approx: X_{\mathcal{D}}^{\mathcal{D}} \rightarrow X_{\mathcal{D}}^{\mathcal{E}}$ such that $s_i^{\mathcal{E}} \in \approx [s_i^{\mathcal{D}}]$ provides a DTaB between $\mathcal{D}$ and $\mathcal{E}$.

4.1. Minimality on Extended MealyDEVS models

For an Extended MealyDEVS model *D*, if *two passive states of D are distinguishable, all the active states to where they can transition will be distinguishable among each other*. Then, for our distinguishing purposes, we consider only passive states

Definition 4.3 (Distinguishing sequence) *Two passive states s_i and s_j of an extended MealyDEVS model D are distinguishable if and only if there exists at least two execution fragments $\alpha = v_{\alpha_0}\ x_{\alpha_1}\ v_{\alpha_1}\ x_{\alpha_2}\ v_{\alpha_2} \ldots x_{\alpha_n}\ v_{\alpha_n}$ and $\beta = v_{\beta_0}\ x_{\beta_1}\ v_{\beta_1}\ x_{\beta_2}\ v_{\beta_2} \ldots x_{\beta_n}\ v_{\beta_n}$ of $\mathcal{M}$ with trace(α) $= (\theta_{I_\alpha}, \theta_{O_\alpha}, t_\alpha)$, trace(β) $= (\theta_{I_\beta}, \theta_{O_\beta}, t_\beta)$ where $v_{\alpha_0}(e) = (s_i, ta(s_i))\ \forall e \in I_{\alpha_0}$, $v_{\beta_0}(e) = (s_j, ta(s_j))\ \forall e \in I_{\beta_0}$, $\theta_{I_\alpha} = \theta_{I_\beta}$ and $(\theta_{O_\alpha} \neq \theta_{O_\beta})$. The timed sequence θ_{I_α} (and θ_{I_β}) is called a distinguishing sequence of the pair (s_i, s_j). If there exists for pair (s_i, s_j) a distinguishing sequence of length k, then the states in (s_i, s_j) are said to be k-distinguishable.*

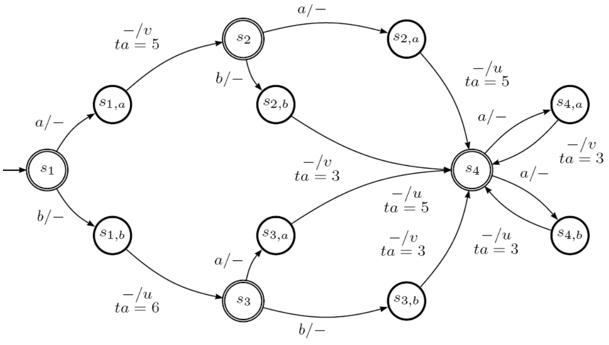
Lemma 4.1 *In the previous definition, θ_{I_α} (and also θ_{I_β}) are slow timed input sequences.*

It should be straightforward to verify that this definition takes into account the discrepancies in the values of *ta* for active states: if $ta(s_{i,x}) \neq ta(s_{j,x})$ for some $s_{i,x}, s_{j,x} \in S_{a\mathcal{D}}$ then they will unavoidably force $s_i, s_j \in S_{p\mathcal{D}}$ to be distinguishable, since the output trace of s_i, $\theta_{O_{si}}$, will have the form $(\theta_{O_{si}} =< \ldots, (x, t_i), \ldots >) \neq (\theta_{O_{sj}} =< \ldots, (x, t_i + \Delta), \ldots >)$, where $\theta_{O_{sj}}$ is the output trace of s_j and $|\Delta| > 0 \wedge (t_i + \Delta) > 0$.

This lemma allows us to define a procedure to be used for *distinguishing* states in Extended MealyDEVS models.

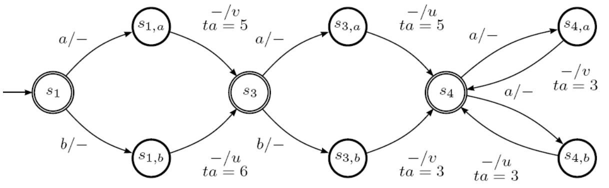
4.1.1. Minimization procedure for Extended MealyDEVS models

A state transition table can be used to represent the functions of an Extended MealyDEVS model, but in this case, we add, in the table, the value of the lifetime of the next active state (Figure **3**). The minimization procedure defined in (Kohavi 1978) is extended in order to take into account the value of $ta(s_i)$ (lifetime of the state s_i).



State	x = a	x = b
s1	s2, 5, v	s3, 6, u
s2	s4, 5, u	s4, 3, v
s3	s4, 5, u	s4, 3, v
s4	s4, 3, v	s4, 3, u

Figure 3: An Extended MealyDEVS model with its associated transition table.



State	x = a	x = b
s1	s3, 5, v	s3, 6, u
s3	s4, 5, u	s4, 3, v
s4	s4, 3, v	s4, 3, u

Figure 4: Resulting minimal Extended MealyDEVS model (c) after the deletion of s_2 (which is equivalent to s_3), $s_{2,a}$ and s_{2_b}, and the resulting transition table (d).

4.2. Fault detection techniques on Extended MealyDEVS models

4.2.1. Experiments on Extended MealyDEVS models

We introduce the concept of a timed *relative* input sequence in order to refer not to the absolute time of each event, but to its relative time with respect to the previous input event. Given a timed input sequence $\pi = \langle(x_0,t_0),(x_1,t_1),\ldots,(x_n,t_n)\rangle$, we refer not to π, but to *rel*(π), where the function *rel* is defined as follows:

$$\begin{aligned} &rel(\langle(x_0,t_0),(x_1,t_1),\ldots,(x_n,t_n)\rangle) \\ &= \langle(x_0,t_0),(x_1,t_1-t_0),\ldots,(x_n,t_n-t_{n-1})\rangle \end{aligned}$$

It is easy to show that *rel* is bijective, so it is equivalent to talk either about π or about *rel*(π) as we can easily define $rel^{-1}(\pi)$. Then, we say that a timed relative input sequence π is *slow* on an Extended MealyDEVS model *D* iff $rel^{-1}(\pi)$ is slow on *D*.

With the previous remark, we will define the experiments for Extended MealyDEVS models:

Definition 4.4 (Preset Experiment) *A timed relative input sequence* $\pi \in (X_{\mathcal{D}} \times \Re_0^+)^*$ *of the form* $\langle(x_0,t_0),..,(x_n,t_n)\rangle$ *defines a preset experiment when an Extended MealyDEVS model D receives it as input. The sequence of output events (the output trace) that D generates in response to* π *is the result of the experiment.*

For example, the following sequence constitutes an experiment when inputted:

$$\pi = \langle(a,0)(b,2)(c,1)\rangle.$$

This experiment is to be interpreted as follows:

> Issue input a; wait 0 units of time; issue input b; wait 2 units of time; issue c; wait 1 unit of time; collect the observed output.

It is straightforward that not all sequences that belong to $(X_{\mathcal{D}} \times \Re_0^+)^*$ can be applied to a given Extended MealyDEVS model. The subset of sequences that can be accepted constitutes the set of valid experiments:

Definition 4.5 (Valid Preset Experiment) *A timed relative input sequence* $\pi \in (X_{\mathcal{D}} \times \Re_0^+)^*$ *of the form* $\langle(x_0,t_0),..,(x_n,t_n)\rangle$ *defines a valid preset experiment when an Extended MealyDEVS model D receives it as input iff* $\forall i = 0..n \bullet t_i > k = max\ \{ta(s) \mid s \in S_{a\mathcal{D}}\}$.

That is to say, a timed input sequence will define a valid experiment if and only if it is slow on *D*.

An *adaptive experiment* is defined as follow:

Definition 4.6 (Adaptive Experiment) *A timed relative input sequence* $\pi \in (X_{\mathcal{D}} \times \Re_0^+)^*$ *of the form* $\langle(x_0,t_0),..,(x_n,t_n)\rangle$ *defines an adaptive experiment when an Extended MealyDEVS model D receives it as input, and provided that there exists a function*

$$f: \quad Seq(X_{\mathcal{D}})\ \times Y_{\mathcal{D}} \rightarrow X_{\mathcal{D}}$$

such that $\forall i = 2..n \bullet \exists y_i \in Y_{\mathcal{D}} \mid x_i = f(\langle x_0,..,x_{i-1}\rangle, y_i)$

That is to say that the value of the i^{th} *input event in* π *depends on all the previous input events and on an (unspecified) output event. The sequence of output events (the output trace) that D generates in response to* π *is the result of the experiment.*

Definition 4.7 (Valid Adaptive Experiment) *An adaptive experiment which consists of inputting the sequence* $\pi = \langle(x_0,t_0),..,(x_n,t_n)\rangle$ *into an Extended MealyDEVS model* $\mathcal{D} = (X_{\mathcal{D}}, Y_{\mathcal{D}}, S_{\mathcal{D}}, \delta_{ext\mathcal{D}}, \delta_{int\mathcal{D}}, \lambda_{\mathcal{D}}, ta_{\mathcal{D}})$ *is considered to be valid if it is slow on D and its function f satisfies the following property:*

- $x_1 = f(\langle x_0\rangle, y_0) \Leftrightarrow y_0 = \lambda_{\mathcal{D}}(\delta_{ext\mathcal{D}}(s_0, e, x_0))$
- $x_i = f(\langle x_0,..,x_{i-1}\rangle, y_{i-1}) \Leftrightarrow$
  $y_{i-1} = \lambda_{\mathcal{D}}(\delta_{ext\mathcal{D}}(f(\langle x_0,..,x_{i-2}\rangle, y_{i-2}), e, x_{i-1}))$

That is to say that the value of the i^{th} input event in π depends on all the previous input events and the last output event that the model generated. This is equal to saying that the i^{th} input event depends on all the output events that the Extended MealyDEVS model has generated so far.

Depending on the results obtained at the end of an experiment, it can be classified into different categories: let *state* be the function defined as:

$$\begin{aligned} &state: \mathbb{N}_0 \rightarrow S_{p\mathcal{D}} \\ &state(0) = s_0 \\ &state(n) = \delta_{int\mathcal{D}}(delta_{ext\mathcal{D}}(state(n-1), e, x_{n-1})) \\ &\qquad \forall n > 0 \end{aligned}$$

then the following categories of experiments are defined:

Definition 4.8 (Distinguishing Experiment) *A valid experiment which consists of inputting the timed relative sequence* $\pi = \langle(x_0,t_0),..,(x_n,t_n)\rangle$ *into an*

Extended MealyDEVS model $\mathcal{D} = (X_{\mathcal{D}}, Y_{\mathcal{D}}, S_{\mathcal{D}}, \delta_{ext\mathcal{D}}, \delta_{int\mathcal{D}}, \lambda_{\mathcal{D}}, ta_{\mathcal{D}})$ *, and whose result is the output sequence* υ*, is distinguishing if there exists an injective function* $first : Seq(X_{\mathcal{D}}) \times Seq(Y_{\mathcal{D}}) \rightarrow S_{p\mathcal{D}}$ *such that* $first(\pi,\upsilon) = state(0) \ \ \forall\upsilon$*.*

Definition 4.9 (Homing Experiment) *A valid experiment which consists of inputting the timed relative sequence* $\pi = \langle(x_0,t_0),..,(x_n,t_n)\rangle$ *into an Extended MealyDEVS model* $\mathcal{D} = (X_{\mathcal{D}}, Y_{\mathcal{D}}, S_{\mathcal{D}}, \delta_{ext\mathcal{D}}, \delta_{int\mathcal{D}}, \lambda_{\mathcal{D}}, ta_{\mathcal{D}})$*, and whose result is the output sequence* υ*, is homing if there exists an injective function* $last : Seq(X_{\mathcal{D}}) \times Seq(Y_{\mathcal{D}}) \rightarrow S_{p\mathcal{D}}$ *such that* $last(\pi,\upsilon) = state(n+1) \ \ \forall\upsilon$*.*

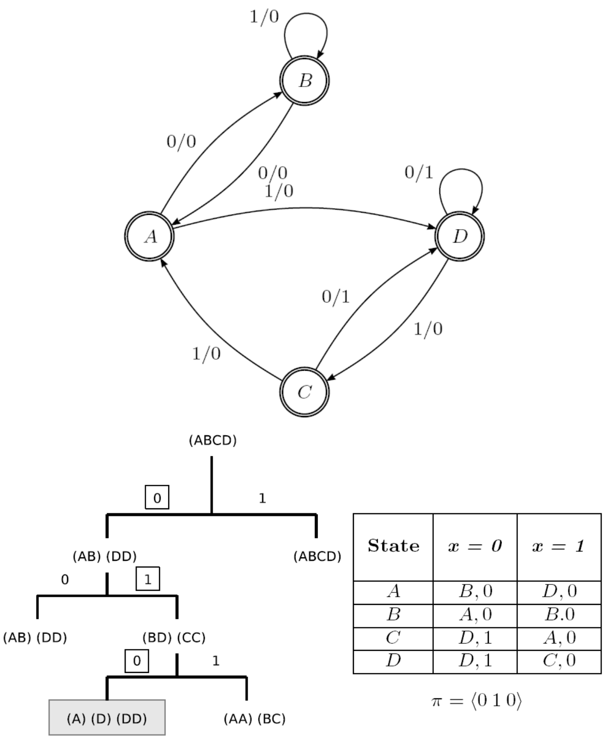
Definition 4.10 (Synchronizing Experiment) *A valid experiment which consists of inputting the timed relative sequence* $\pi = \langle(x_0,t_0),..,(x_n,t_n)\rangle$ *into an Extended MealyDEVS model* $\mathcal{D} = (X_{\mathcal{D}}, Y_{\mathcal{D}}, S_{\mathcal{D}}, \delta_{ext\mathcal{D}}, \delta_{int\mathcal{D}}, \lambda_{\mathcal{D}}, ta_{\mathcal{D}})$ *, and whose result is the output sequence* υ*, is synchronizing for state* $s_k \in S_{p\mathcal{D}}$ *if there exists a function* $last : Seq(X_{\mathcal{D}}) \times Seq(Y_{\mathcal{D}}) \rightarrow S_{p\mathcal{D}}$ *such that* $last(\pi,\upsilon) = s_k \ \ \forall\upsilon$.

4.2.2. Sequence Finding

In order to perform a *homing experiment* for an Extended MealyDEVS model, the procedure described in (Kohavi 1978) can be utilized in a straightforward way, by considering the uncertainty of the model to be composed of all passive states in it.

By applying the algorithm given in (Kohavi 1978), a preset homing sequence can be designed for minimal Extended MealyDEVS models by adjoining, to each input event, a value $t_k > t = max \ \{ta(s) \ \ | s \in S_{a\mathcal{D}}\}$. Thus, we extend the requirements on the models to be tested (Kohavi 1978, Lee and Yannakakis 1996) so as to have *t* as a given value.

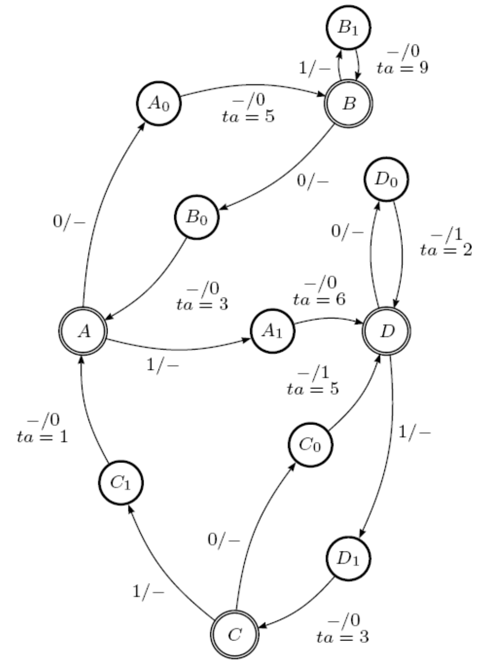
As an example, we give a Mealy machine *M*, its associated homing tree, and the resulting homing sequence for this machine (Figure 5). Secondly, we give (Figure 6) an Extended MealyDEVS model (which has a delay time-abstracting bisimulation relation with *M)*, and the resulting homing sequence (note that $k = max\{ta_{\mathcal{D}}(s_{a_i}) \,|\, s_{a_i} \in S_{a\mathcal{D}}\} = 9$. Then, every waiting value after an input event should be > 9).

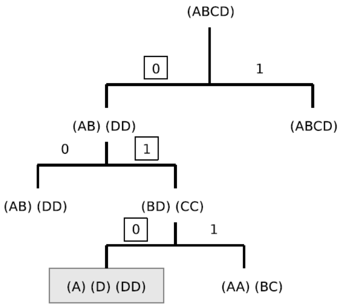


State	$x = 0$	$x = 1$
A	$B, 0$	$D, 0$
B	$A, 0$	$B.0$
C	$D, 1$	$A, 0$
D	$D, 1$	$C, 0$

$\pi = \langle 0\ 1\ 0 \rangle$

Figure 5: A Mealy machine with its homing tree, associated transition table, and a homing sequence π for it.





State	$x = 0$	$x = 1$
A	$B, 0, 5$	$D, 0, 6$
B	$A, 0, 3$	$B, 0, 9$
C	$D, 1, 5$	$A, 0, 1$
D	$D, 1, 2$	$C, 0, 3$

$$\pi = \langle (0, 9.2)\ (1, 10)\ (0, 12) \rangle$$

Figure 6: Extended MealyDEVS model derived from the one in Figure 5 with its homing tree, associated transition table, and one possible homing sequence (π) for it.

Given the previous considerations on the timed nature of sequences, both *distinguishing* and *synchronizing* sequences can be obtained by using the methods and algorithms described in (Kohavi 1978, Lee and Yannakakis 1996) and afterwards adjoining the time components as was previously described in this section. It is straightforward to prove that all the properties and theorems are valid for Extended MealyDEVS models. In particular, the following results are valid also for Extended MealyDEVS models:

Theorem 4.2: *A preset homing sequence, whose length is at most* $(n-1)^2$, *exists for every minimal Extended MealyDEVS model* $\mathcal{D}$ *, where* n *is the number of passive states in* $\mathcal{D}$.

Theorem 4.3: *If there exists a synchronizing sequence for an Extended MealyDEVS model D that has n passive states, then its length is at most* $(n-1)^2 n/2$.

Proof: *See (Kohavi 1978).*

The following result sums up the preceding discussion:

Theorem 4.4: *Let* $\pi = \langle x_1 x_2 .. x_n \rangle$ *be either a synchronizing, homing, preset distinguishing or adaptive distinguishing sequence for a Mealy Machine M. Then the sequence* $\upsilon = \langle (x_1, t_1)(x_2, t_2)..(x_n, t_n) \rangle$ *obtained from* π *is, respectively, either a synchronizing, homing, preset distinguishing or adaptive distinguishing sequence for the Extended MealyDEVS model D obtained from M if* $t = max \{ta_{\mathcal{D}}(p) \mid p \in S_{a\mathcal{D}}\} < t_i, \forall i = 1..n$**.**

4.2.3. Testing procedure

In order to be able to represent the behaviour defined by the semantics of a timed sequence, it is necessary to model a *tester* (Krichen and Tripakis 2005), that is, a DEVS model that emits the events (at a specified time) needed to test a given Extended MealyDEVS model.

The general scheme of the coupling between a tester and the model to be tested is:

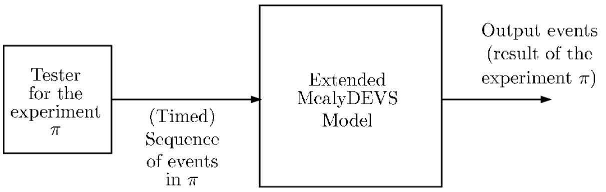

Tester for the experiment π
(Timed) Sequence of events in π
Extended MealyDEVS Model
Output events (result of the experiment π)

Figure 7: Coupling scheme of an Extended MealyDEVS model and a valid tester for it.

And the tester for a given PX is defined as follows:

Definition 4.11 (DEVS tester model) *Given a PX* $\pi = (x_0, t_0), (x_1, t_1), \ldots, (x_n, t_n)$, *its associated tester is the DEVS model* $\mathcal{T} = < X_{\mathcal{T}}, Y_{\mathcal{T}}, S_{\mathcal{T}}, \delta_{int_{\mathcal{T}}}, \delta_{ext_{\mathcal{T}}}, \lambda_{\mathcal{T}}, ta_{\mathcal{T}} >$, *where:*

- $X_{\mathcal{T}} = \{Reset\}$ *(Event that restores the tester to its initial state)*
- $Y_{\mathcal{T}} = x_0, x_1, \ldots, x_n$
- $S_{\mathcal{T}} = s_0, s_1, \ldots, s_n \cup s_{STOP}$
- $\delta_{ext_{\mathcal{T}}}(s_i, e, x) = s_0$ *(Restores the tester to its initial state)*
- $\delta_{int_{\mathcal{T}}}(s_i) = \begin{cases} s_{i+1} & \text{if } i < n \\ s_{STOP} & \text{if } i = n \vee s_i = s_{STOP} \end{cases}$
- $ta_{\mathcal{T}}(s_i) = \begin{cases} 0 & \text{if } i = 0 \\ t_{i-1} & \text{if } i = 1..n \wedge s_i \neq s_{STOP} \\ \infty & \text{if } s_i = s_{STOP} \end{cases}$
- $\lambda_{\mathcal{T}}(s_i) = \begin{cases} x_i & \text{if } s_i \neq s_{STOP} \\ \varnothing & \text{if } s_i = s_{STOP} \end{cases}$

  (*The case* $\lambda_{\mathcal{T}}(s_i) = \varnothing$ *never happens**,***
  as $ta_T(s_{STOP}) = \infty$***)***

As an example, the tester that implements the experiment $\pi = (a,0)(b,2)(c,1)$ is given in figure 8.

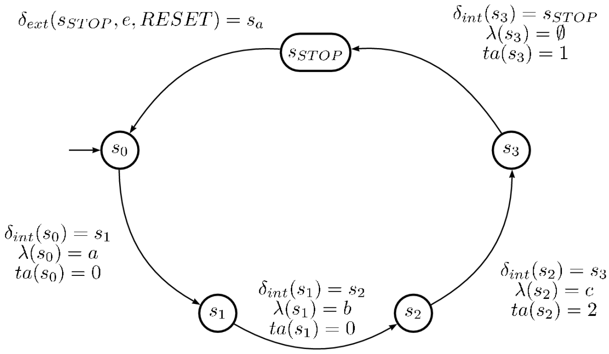

$\delta_{ext}(s_{STOP}, e, RESET) = s_a$
$\delta_{int}(s_3) = s_{STOP}$
$\lambda(s_3) = \emptyset$
$ta(s_3) = 1$
s_{STOP}
s_0
s_3
$\delta_{int}(s_0) = s_1$
$\lambda(s_0) = a$
$ta(s_0) = 0$
$\delta_{int}(s_2) = s_3$
$\lambda(s_2) = c$
$ta(s_2) = 2$
$\delta_{int}(s_1) = s_2$
$\lambda(s_1) = b$
$ta(s_1) = 0$
s_1
s_2

Figure 8: Tester for the preset experiment (PX) $\pi = (a,0)(b,2)(c,1)$.

If we take into account the fact that after the tester sends the output event x_i, the Extended MealyDEVS model under test will transition up to the passive state s_{i+1}, then it is assured that the Extended MealyDEVS model will remain in the same state until the tester issues the output event x_{i+1} (in case it exists).

In order to implement a valid adaptive experiment for an Extended MealyDEVS model, the basic idea is to define a tester which represents the decision tree associated with the adaptive experiment (Kohavi 1978). It is then required to have one active state for each node in the tree (that is, one node for each possible uncertainty in the tree). Each such state will output the event that the Extended MealyDEVS model under test is going to receive in order to solve the uncertainty. Additionally, each of these states will (internally) transition to a passive state that represents the response of the model under test. That is, it will be able to receive any of the possible output events that the Extended MealyDEVS model will generate (in order to do this, the output of the tested model needs to be connected to the tester's input). Depending on the value of the received event, the tester will transition to one of the active states that represent the consequential uncertainties (the nodes one that are one level lower). Finally, all the leaf nodes have to be represented as passive states which only accept the RESET event in order to reinitialize the experiment. The following figure shows a concrete example of an adaptive tester:

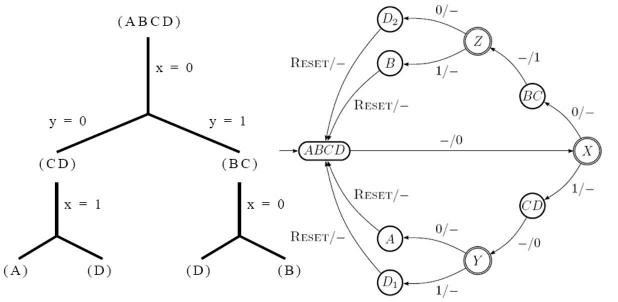


Figure 9: Sample adaptive experiment and corresponding DEVS tester model.

5. CONCLUSION

In this paper, we have proposed a first approach, on a subset of DEVS models, on which we can apply extensions of fault detection techniques from Mealy machines.

For the given subset of DEVS models to which the theory of black-box testing can be applied, we have proposed:

- a proper extension and formalization of the concepts of homing, distinguishing and synchronizing sequences,
- the preset and adaptive experiments that the three previously mentioned types of sequences allow for.

Finally, we have given the procedure and structure of the DEVS models that implement the testing procedure for both preset and adaptive fault detection experiments.

Further lines of work involve expanding the proposed approach to more general DEVS and to define the limitations of these techniques in a timed formalism as DEVS in order to clearly define the broadest possible subset of DEVS models which can be black-box tested.

REFERENCES

Dacharry, H. P. and Giambiasi, N., 2005. Formal verification with timed automata and DEVS models: a case study. *ASSE 2005 Simposio Argentino de Ingeniería de Software - 34 JAAIO Jornadas Argentinas de Informatica e Investigacion Operativa*, pp. 251–265. August 24 – September 2, Rosario (Santa Fe, Argentina).

Dacharry, H.P. and Giambiasi, N., 2007. DEVS based timed hierarchy of formalisms. *Proceedings of the International Modeling and Simulation Multiconference 2007 (IMSM07)*. SCS -- The Society for Modeling and Simulation International. February 8-10. Buenos Aires (Argentina).

Giambiasi, N., Dacharry, H.P., 2007. DEVS and Timed Automata for the Design of Control Systems. *Robotics Research Trends*, Xing P. Guo (Ed.), Nova Science Publishers, ch. 5, pp. 193-222.

Kohavi, Z., 1978. *Switching and Finite Automata Theory: Computer Science Series*. McGraw-Hill Higher Education.

Krichen, M. and Tripakis, S., 2005. State identification problems for timed automata. In *The 17th IFIP Intl. Conf. on Testing of Communicating Systems (TestCom'05)*, volume 3502 of *LNCS*. Springer.

Lee, D. and Yannakakis M., 1996. Principles and methods of testing finite state machines - A survey. *Proceedings of the IEEE*, volume 84, pages 1090–1126.

Lynch, N. A. and Vaandrager, F. W., 1993. Forward and backward simulations – part I: untimed systems. *Technical Report: CS-R9313*, page 35. Centrum voor Wiskunde en Informatica (CWI).

Lynch, N. A. and Vaandrager, F. W., 1993. Forward and backward simulations – part II: timing-based systems. *Technical Report: CS-R9314*, page 36. Centrum voor Wiskunde en Informatica (CWI).

Springintveld, J, Vaandrager, F. and D'Argenio, P. R., 2001. Testing timed automata. *Theoretical Computer Science*, 254(1–2):225–257.

Tripakis, S., and Yovine, S., 2001. Analysis of timed systems using time-abstracting bisimulations. *Formal Methods in System Design*, 18(1):25–68.

Zeigler B. P., Praehofer, H. and Kim, T. G., 2000. *Theory of Modeling and Simulation, Second Edition*. Academic Press.

Analysis of Shipboard Survivable Fire Main Systems

Albert Ortiz[1], Don Dalessandro[1], Dong Qing[1], Li Bai[2] and Saroj Biswas[2]

1. Department of the Navy, Naval Surface Warfare Center, Carderock Division, Philadelphia, USA
2. Department of Electrical and Computer Engineering, Temple University, Philadelphia, USA

ABSTRACT

This paper presents a study for the shipboard fire main systems using a new probabilistic approach to analyze the survivability of a system. Similarities and differences between survivability and reliability analysis are compared. In a reliability model, one can describe a k-out-of-n:G system (k <= n), in which the component system is valid only if any k or more components function. The system can also be configured into an initial k-out-of-n:G model with m backup components (0 <= m <= n). If the system cannot perform its intended function, the m backup components will be reconfigured with the remaining working components into a new form to sustain system function. Academia refers to such studies to calculate the system successful probabilities as the survivability analysis. In this paper, we focus on the survivability analysis of a shipboard fire main piping system. This study could potentially be used to analyze the survivability of power network systems, dependable secure computing systems, military reconfigurable information systems, and other large reconfigurable network systems.

Keywords – survivability, reliability, k-out-of-n:G system, reconfiguration.

1. INTRODUCTION

In shipboard applications, many systems are built with reconfiguration capabilities that the systems can still perform its intended operations when the initial configuration cannot sustain its original functions due to break down of certain components or sub-components. In the theory of evolution, Charles Darwin addressed the issue of reconfiguration capability as a way to survive through species' adaptation due to environment changes. He referred natural selection or survival of the fittest as a direct result from variations. To complement the theory of reliability defined for redundancy systems, Bai et. al. [1] proposed a probabilistic definition of survivability and developed a survivability framework for redundancy systems which are capable of reconfiguring themselves at the event of failure. Since the system can have different configurations, situation assessment becomes a direct extension of the theory because the system has to correctly identify the direction of threat.

Papanikolaou and Boulougouris [2] also addressed design aspects of survivability for surface naval and merchant ships. They offered a mathematical formula showing how to compute survivability. According to their definition, the survivability P_s is calculated as

$$P_s = 1 - P_k = 1 - P_{su}P_v \,,$$

where P_k is the killability, P_{su} is the susceptibility and P_v is the vulnerability. This formulation is very intuitive and a top-down approach. Since large integrated system often consists of many subsystems such as power modules, communication modules, or computation modules, etc, they are not only susceptible for failure in the direction of threats, also are affected by cascade failures from other interconnected subsystems. To calculate system's survivability, the system can first be divided into these smaller subsystems in terms of probability of susceptibility. The vulnerability of the system is directly dependent on the reliability of subsystems. As a result, the survivability formulation for a reconfigurable system becomes complex and difficult to be evaluated. Varshney et al. [3] explained the difference between reliability and survivability in the context of mean time between failures (MTBF). Likewise, the definition did not consider the system reconfiguration.

In this paper, we review the theory of survivability proposed by Bai et. al. [1] using a simple reconfigurable piping system and investigate the survivability of the fire main systems. The rest of the paper is organized as follows. The main survivability ideas are reviewed and presented in section II. Section III is given to show how the survivability is calculated for the fire main systems. We conclude the paper in section IV.

2. REVIEW OF SURVIVABILITY

To better understand the reliability and survivability analysis, we describe briefly several key reliability models.

2.1 Reliability Models

Typically, a reliable system has redundant components to sustain system function if a few components fail. For example, there are various studies on *k*-out-of-*n*:G or *k*-out-of-*n*:F systems. Kuo and Zuo [4] classified:

i. The *k*-out-of-*n*:G system works (well) when at least k components work among all n components.
ii. The *k*-out-of-*n*:F system fails when at least k components can not function among all n components.

These two systems are equivalent where a *k*-out-of-*n*:G system is the same as a (*n*–*k*+1)-out-of-*n*:F system. The reliability of a *k*-out-of-*n*:G system is to compute the successful probability of the system. For example, we can calculate the reliability of a *k*-out-of-*n*:G with *n* identical components whose successful probabilities are p as,

$$R = \sum_{i=k}^{n} \binom{n}{i} p^{i} (1-p)^{n-i} ,$$

where $\binom{n}{i}$ is the combination for *n* choose *i*. Also, the reliability model considers the uniform threat from all directions. In other words, a specific threat direction does not have any influence on the successful probability of the components in the reliability model, or the reliability stays the same regardless of where the threat is from.

2.2 Survivability Models

A survivable system is a reliable system with reconfiguration capability. To precisely define survivability, the initial form (or original configuration) is an important factor which is also directly related to where the threat direction is from. The system can perform its functions by varying into a new form when it cannot survive in its original form. We define the survivability of a reconfigurable system as

$$S = R(f_0) + \sum_{i \neq 0} Q(f_0 \rightarrow f_i) A(f_i) R(f_i, c_i) \qquad (1)$$

where $R(f_0)$ is the reliability of an initial configuration f_0 and $A(f_i)$ is probability of successful adaptation into a new configuration f_i. Since a system has to be fault tolerant, the configuration f_0 requires several redundant components in order to provide sufficient reliability. When threats come from different directions, we can compute reliability of each component by using total probability theorem as,

$$p(c) = \sum_{j} p(c \mid T_j) p(T_j)$$

where T_j is the direction of a threat, p(T_j) is a-prior probability of the threat, and $p(c|T_j\ )$ is conditional reliability of component based on a particular threat T_j . The component reliability can further be classified as a *k*-out-of-*n*:G (good) system reliability metric shown as $R(f_0)$ and $R(f_i,\ c_i)$. This formulation includes the idea of susceptibility, reliability as well as adaptability for possible reconfiguration solutions. As shown in the formulation, there is a term $Q(f_0 \rightarrow f_i)$ that implies the system requires modification from its initial form. Since the modification can occur under different circumstances, it can result in the following two types of survivability analysis: i) adaptation and ii) mutation.

i. Adaptation survivability refers to a system that reconfigures itself only when its initial form fails to work.
ii. Mutation survivability refers to a system that can reconfigure itself even when its initial form is still performing its tasks.

In our engineering analysis of survivability, we simply investigate the adaptation survivability of a system because many survivable engineering applications require a new configuration to sustain operations only when the initial form fails to work. During reconfiguration states, engineers and technicians can be dispatched to repair the failed components in their initial form. It is apparent that adaptation survivability is a more applicable analysis for engineering survivable systems.

2.3 A Simple Survivable System

First, we consider a simple survivable system shown in Figure 1 where the system has two pumps and only one can be operated due to a limited power supply or pipe pressure. The pump supplies enough water to three sprinkler pipes in the middle segment of the system at any given time. We refer to the middle segment as the survivable space. As we can see, the main threat is from the right hand-side of the system and we can assume that $p_1 < p_2$ where p_i is the successful probability of the pump *i*. However, a reliable model will not consider the two pumps as being different because the system is unaware of where the threat is coming from. Rather, an initial form can be chosen to either operate pump 1 or operate pump 2.

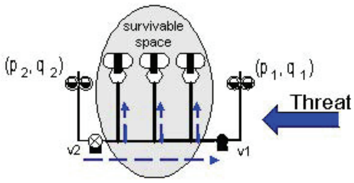


Fig. 1. Two Pumps Shipboard Firemain System

After the initial form is chosen to operate pump 1, the reliability is

$$R = p_1,$$

or the reliability can become p_2 if pump 2 is operated in its initial form.

The values of p_1 and p_2 are highly dependant on where the threat is coming from. If a reliable and reconfigurable system can identify where the threat is coming from and reconfigure itself accordingly, the system becomes survivable. A better way is to operate the pump with the lower threat. In other words, the appropriate survivable system is to operate pump 2 and reconfigure itself to operate pump 1 when pump 2 fails. Here we have a survivable model with an initial form of operating a 1-out-of-1:G system for pump 2 with pump 1 as a backup. If pump 2 fails, there are four reconfiguration possibilities to open or close valve 1 and valve 2. Among them, two possibilities enable pump 1 to supply water flow to the sprinkler pipes as shown in Figure 2:

i. valve 1 is on and valve 2 is on,
ii. valve 1 is on and valve 2 is off.

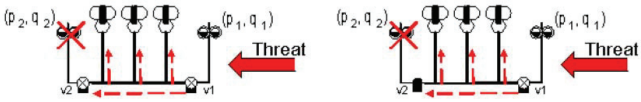


Fig. 2. Pumps Reconfiguration Choices

Consequently, the system survivability is computed by using (1) as

$$S = R(f_0) + \sum_{i=1}^{2} Q(f_0 \to f_i) A(f_i) R(f_i, P^{(1)})$$

$$= p_2 + q_2 \frac{2}{4} p_1 = p_2 + \frac{1}{2} q_2 p_1,$$

where q_2 is the failure probability of pump 2. Conversely, we can define the non-survivability as

$$\overline{S} = \sum Q(f_0 \to f_i)(A(f_i)Q(f_i, c_i) + \overline{A}(f_i)) \qquad (2)$$

where $A(f_i)$ is the probability of successfully adaptation, $\overline{A}(f_i)$ is the probability of failed adaptation, and $Q(f_i, c_i)$ is the failure probability of newly added components c_i in the newly reconfigured form f_i. The definition is relatively easy to understand that

i. The term $Q(f_0 \to f_i)$ indicates the probability that a new form f_i will be reconfigured.
ii. The term $A(f_i)Q(f_i, c_i)$ indicates that newly components c_i fails to work in the new form f_i.
iii. The third term $\overline{A}(f_i)$ indicates that the system cannot be updated into a new form f_i.

These conditions all produce a system without survivable options. We can use the same idea to compute the non-survivability of the current system when pump 2 fails to work. There are two situations:

i. valve 1 is on but pump 1 fails, and
ii. valve 1 cannot be turned on.

In both conditions, we can use (2) to calculate the non-survivability as,

$$\overline{S} = q_2 \left(\frac{1}{2} q_1 + \frac{1}{2} \right)$$

Interesting enough, we can also verify that the sum of the survivability and non-survivability is unity, or

$$S + \overline{S} = p_2 + \frac{1}{2} q_2 p_1 + q_2 \left(\frac{1}{2} q_1 + \frac{1}{2} \right) = 1 .$$

In the current system, we can easily see that S > R. It implies that a survivable system can have a higher successful probability than a reliable system. The survivable system is capable of configuring an initial form depending on where the threat is coming from, and it can reconfigure itself to avoid failures. If the system cannot identify where the threat is coming from, its survivability will be degraded. For the same system shown in Figure 1, the survivability of the system when it can identify the threat correctly is

$$S = p_2 + \frac{1}{2} q_2 p_1 .$$

We can compare it with another system that identifies the threat as coming from the wrong direction, or the system operates pump 1 in its initial form. The survivability of such a system is

$$\widetilde{S} = p_1 + \frac{1}{2} q_1 p_2 .$$

Clearly, we can calculate

$$S - \widetilde{S} = \left(p_2 + \frac{1}{2} q_2 p_1 \right) - \left(p_1 + \frac{1}{2} q_1 p_2 \right) = \frac{3}{2} (p_2 - p_1) \ge 0 .$$

More precisely, we prove that $S \ge \widetilde{S}$. This result suggests that a reconfigurable system is more survivable if its initial form is determined by avoiding the threat. In other words, a threat aware and reconfigurable system has a clear advantage in terms of better survivability. We can prove this concept in the following theorem.

Theorem 1: Survival of the Fittest – *A threat aware and reconfigurable system is more capable of surviving.*

Proof For a more general system, suppose we have two different initial forms, $f_0^{(1)}$ and $f_0^{(2)}$. If the form $f_0^{(1)}$ is configured with threat awareness capability, we have $R\left(f_0^{(1)}\right) > R\left(f_0^{(2)}\right)$. Since both initial forms have the same number of components and backup components. We have same number of survivable forms available for the survivable options, or

$$R\left(f_0^{(1)}\right) + \sum_{i \ne 0} Q(f_0^{(1)} \to f_i) = 1,$$

$$R\left(f_0^{(2)}\right) + \sum_{i \ne 0} Q(f_0^{(2)} \to f_i) = 1.$$

Since $R\left(f_0^{(1)}\right) > R\left(f_0^{(2)}\right)$, it implies that

$$\sum_{i \ne 0} Q(f_0^{(1)} \to f_i) < \sum_{i \ne 0} Q(f_0^{(2)} \to f_i), \text{ or}$$

$$\sum_{i \ne 0} Q(f_0^{(1)} \to f_i) - \sum_{i \ne 0} Q(f_0^{(2)} \to f_i) < 0..$$

Also, since $A(f_i) \ge 0$ and $Q(f_i, c_i) \ge 0$, we know $A(f_i)Q(f_i, c_i) + \overline{A}(f_i) \ge 0$. It suggests that

$$\sum_{i \ne 0} Q(f_0^{(1)} \to f_i)\left(A(f_i)Q(f_i, c_i) + \overline{A}(f_i)\right) - \sum_{i \ne 0} Q(f_0^{(2)} \to f_i)\left(A(f_i)Q(f_i, c_i) + \overline{A}(f_i)\right) < 0.$$

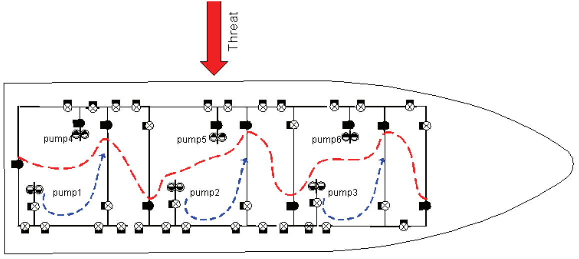
It implies that $\overline{S}^{(1)} - \overline{S}^{(2)} < 0$. We know that $S = 1 - \overline{S}$, it means that $S^{(1)} > S^{(2)}$. ■

From Theorem 1, we validate the concept that a reconfigurable system has better survivability if the system has threat awareness capability. Nonetheless, we demonstrate the difference between the survivability and reliability analysis.

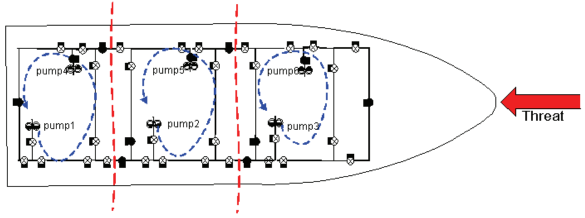
3. SURVIVABILITY OF SHIPBOARD FIRE MAIN SYSTEM

In a simplistic view of the shipboard fire main system, there are six pumps, and each pump can supply water flow into the sprinkler pipes. There are two types of threats that can result with two initial configurations as shown in Figures 3(a) and 3(b).

In either configuration for the complete shipboard survivability, maximum three pumps are allowed to operate in the same time, say that pumps 1-3 are in operation to supply the water flow into the sprinkler pipes so that water pressure will be provided to extinguish fires. Generally, at least one pump per compartment or two pumps per three compartments have be operated so that there is enough water pressures for extinguishing fire.



(a) side configuration



(b) zone configuration

Consequentially, system design allows the following two initial configurations:

- side configuration (shown in Figure 3(a)) has an initial 2-out-of-3:G system (two pumps per three compartment). When any two pumps fail, the system will reconfigure and sustain the survivable mission.
- zonal configuration (shown in Figure 3(b)) has three initial 1-out-of-1:G systems (one pumps per compartment). When a pump in any subsections fails, other pump will be in operation to sustain the survivable mission.

Here, we denote $p^{(i)}$ as successful probability and $q^{(i)}$ as failure probability of the i-th pump respectively. Then, we can determine the survivability two shipboard fire main system configurations.

3.1 Survivability of Side Configuration

As shown in the Figure 3(a), the threat comes from the opposite side of pumps 1-3. Therefore, we can assume

$$p^{(1)} = p^{(2)} = p^{(3)},\ p^{(4)} = p^{(5)} = p^{(6)},$$
$$q^{(1)} = q^{(2)} = q^{(3)} \text{, and } q^{(4)} = q^{(5)} = q^{(6)}.$$

For simplicity, we denote

$$p^{(1)} = p^{(2)} = p^{(3)} = p1,$$
$$p^{(4)} = p^{(5)} = p^{(6)} = p2,$$
$$q^{(1)} = q^{(2)} = q^{(3)} = q1,$$
$$q^{(4)} = q^{(5)} = q^{(6)} = q2.$$

Also, $p1+q1 = 1$ and $p2+q2 = 1$. Since the threat is from the side of pumps 4-6, we know that $p1 > p2$. The initial configuration is to allow pumps 1-3 to operate in a 2-out-of-3:G system because the system can provide at least two pumps' water pressure to three compartments. Therefore, the reliability of the initial system configuration is

$$R(f_0) = p_1^3 + \binom{3}{2} p_1^2 q_1$$

There are two conditions that the system must be reconfigured.

1. When any two pumps among pumps 1-3 fail, any one or two pumps among pumps 4-6 are switched open to continue operating with the pumps remain working.
2. When all three pumps among pumps 1-3 fail, the section of the pumps are close. Any two or three pumps among pumps 4-6 have to be switched open.

Therefore, the probability that the first condition will occur with either one or two pumps and corresponding valve opening correctly as

$$\binom{3}{1}\frac{1}{8}p_2 + \binom{3}{2}\frac{1}{4}p_2^2,$$

Similarly, the probability is the second condition is

$$\binom{3}{2}\frac{1}{4}p_2^2 + \binom{3}{3}\frac{1}{8}p_2^3,$$

The system survivability is

$$S_{side} = R(f_0) + \sum_{i \neq 0} Q(f_0 \to f_i) A(f_i) R(f_i, c_i)$$

$$= p_1^3 + \binom{3}{2} p_1^2 q_1 + \binom{3}{2} q_1^2 p_1 \left[\binom{3}{1}\frac{1}{8}p_2 + \binom{3}{2}\frac{1}{4}p_2^2 \right] + \binom{3}{3} q_1^3 \left[\binom{3}{2}\frac{1}{4}p_2^2 + \binom{3}{3}\frac{1}{8}p_2^3 \right]$$

3.2 Survivability of Zonal Configuration

As shown in the zone configuration in Figure 3(b), the successful and failure probabilities of pumps are different than that in the side configuration because the threat is in the front of the ship. The probabilities are:

$$p^{(1)} = p^{(4)},\ p^{(2)} = p^{(5)},\ p^{(3)} = p^{(6)},$$
$$q^{(1)} = q^{(4)},\ q^{(2)} = q^{(5)}, \text{ and } q^{(3)} = q^{(6)}.$$

Also, we denote

$$p^{(1)} = p^{(4)} = p1,$$
$$p^{(2)} = p^{(5)} = p2,$$

$$p^{(3)} = p^{(6)} = p3,$$
$$q^{(1)} = p^{(4)} = q1,$$
$$q^{(2)} = p^{(5)} = q2,$$
$$q^{(3)} = p^{(6)} = q3.$$

Also, $p1+q1 = 1$, $p2+q2 = 1$, and $p3+q3 = 1$. Since the threat is coming from in the front of the ship, we know that $p1 > p2 > p3$. The initial configuration is to allow pumps 1-3 to operate in a 1-out-of-1:G system because the system can provide at least two pumps' water pressure to three compartments.

There are many possibilities that different zone can survive from the attack and each compartment has the exact same survivability, we can compute the non-survivability of one zone before we calculate whole shipboard survivability.

To compute the non-survivability of the compartment with pumps 1 and 4, there are two possibilities that the subsection can fail completely:

1. When pump 1 fails, pump 4 is not operable even the pump valve is switched open, and
2. When pump valve cannot be opened despite whether pump 4 is operable.

Therefore, the non-survivability of the subsection can be calculated as

$$\bar{S}_1 = Q(f_0 \to f_1)\left(\frac{1}{2}q_1 + \frac{1}{2}\right) = \frac{1}{2}q_1(q_1+1).$$

Consequentially, the whole shipboard system survivability is the product of three subsections' survivability

$$S_{zone} = \prod_{i=1}^{3}(1-\bar{S}_i) = \prod_{i=1}^{3}\left(1-\frac{1}{2}q_i(q_i-1)\right).$$

3.3 Wrong Fire Main Configuration

Intuitively, zonal configuration is suitable for the threat is coming from in the front of ship; and side configurable is better for the threat is from the side of the ship. Here we demonstrate how to use the survivability metric why the zonal configurable cannot be used when the threat is coming from the side of the ship as shown in Figure 4.

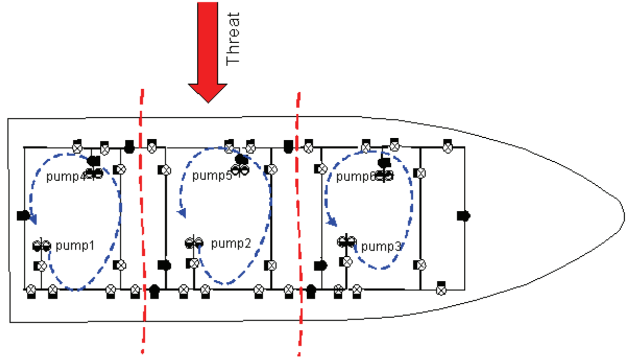


Figure 4. Wrong Fire Main Configuration

As shown in the Figure 4, the threat comes from the opposite side of pumps 1-3. We know that

$$p^{(1)} = p^{(2)} = p^{(3)},\ p^{(4)} = p^{(5)} = p^{(6)},$$
$$q^{(1)} = q^{(2)} = q^{(3)},\ \text{and}\ q^{(4)} = q^{(5)} = q^{(6)}.$$

We denote

$$p^{(1)} = p^{(2)} = p^{(3)} = p1,$$
$$p^{(4)} = p^{(5)} = p^{(6)} = p2,$$
$$q^{(1)} = q^{(2)} = q^{(3)} = q1,$$
$$q^{(4)} = q^{(5)} = q^{(6)} = q2.$$

Also, $p1+q1 = 1$ and $p2+q2 = 1$. Since the threat is from the side of pumps 4-6, we know that $p1 > p2$. However, the initial configuration is to allow pumps 1-3 to operate in a 1-out-of-1:G system because it is in the zonal configuration.

Using the same analysis from the zonal configuration, we can find the whole shipboard survivability as

$$S_{wrong} = \prod_{i=1}^{3}(1-\bar{S}_{wrong}) = \left(1-\frac{1}{2}q_1(q_2+1)\right)^3$$
$$= \left(p_1 + \frac{1}{2}q_1p_2\right)^3.$$

One can easily validate that $S_{side} > S_{wrong}$. It implies that the side configuration should be used when the threat is coming in the side of the ship. The similar result can be obtained if we use side configuration for the threat which is coming the front of the ship. Consequentially, situation awareness is important for the survivability of the shipboard fire main system.

4. CONCLUSION

In this paper, we present a theory for analyzing survivability, and show how it is different from reliability analysis. We are able to use the theory to compute the survivability of shipboard fire main systems. Furthermore, we can mathematically describe how effective a configuration can be measured to survive threat.

REFERENCES

[1] Li Bai, Saroj Biswas, Albert Ortiz, Frank Ferrese, Don Dalessandro and Qing Dong, "Survivability Analysis of Reconfigurable Systems", the International Conference on Industrial Engineering and Engineering Management (IEEM) 2007, Singapore.

[2] Papanikolaou, A., Boulougouris, E. "Design aspects of survivability of surface naval and merchant ships", in Proceedings of International Conference on Naval Technology, Piraeus, Greece (1998)

[3] Varshney, U., Snow, A.P., Malloy, A.D. "Measuring the reliability and survivability of infrastructure-oriented

wireless networks" in The 26th Annual IEEE Conference on Local Computer Networks, 2001. Proceedings. LCN 2001, IEEE (2001) pp. 611 – 618

[4] Kuo, W., Zuo, M.J. "Optimal Reliability Modeling: Principles and Applications". John Wiley & Sons, Inc. (2002), chapter 7, pp. 231–280.

MODELING AND EVALUATION OF DISTINCT ALTERNATIVE DESIGNS FOR WIDE-BAND AIR-COUPLED PIEZOELECTRIC TRANSDUCERS

A. Ruíz [(a)], T. E. Gómez [(a)]

[(a)] *Dept. Ultrasonic Signals, Systems and Technology*
Institute of Acoustics (CSIC)
Serrano 144, 28006 MADRID, (SPAIN)

[(a)] art@ia.cetef.csic.es, tgomez@ia.cetef.csic.es

ABSTRACT

Air-coupled ultrasonic piezoelectric transducers have demonstrated their usefulness in several areas such as materials characterization, surface metrology, or NDE. Although sensitivity is good, their bandwidth is quite narrow. This can be improved by adding two or more matching layers. Usually, these layers become unpractical because they are either extremely thin, or light or brittle and such transducers have to be constructed using materials that are far from the ideal specifications. Here, a comprehensive modeling tool (PSPICE) for the design of air-coupled piezoelectric transducers with improved bandwidth, considering real and low attenuating materials is presented. Several characteristic broadband functions, were simulated in time-frequency domains, for selected sets of the piezoelectric and matching layer materials. Finally, the best transducer design considering real materials within the double-matching layer scheme is proposed. The analysis is restricted to transducers having a centre frequency at 1 MHz. The simulated transducer performance is compared with experimental data.

Keywords: air-coupled transducer, matching layer, PSPICE, material properties.

1. INTRODUCTION

The application of air-coupled ultrasonic devices have achieved a notable impetus during the last years. This has been demonstrated in different areas such as environment protection, health care or Non Destructive Testing (see Grandia and Fortunko 1995 for an early review) .

Air-coupled piezoelectric transducers show a good sensitivity, but their bandwidth is narrower compared with electrostatic devices (Hutchins and Schindel 1994). An important application of air-coupled ultrasounds is for the study and characterization of materials that can not be wetted (as it is usually done in conventional ultrasonic techniques for they require the use of coupling fluids between transducer and samples) (Fortunko, Renken and Murray, 1990, McIntyre, Hutchins, Billson and Stor-Pellinen, 2001, Gómez 2002, Gómez 2003a, Gómez 2003b, Gómez, Apel and Orelovitch 2007). It will be an important advantage, especially for these spectroscopic applications if present bandwidth can be further improved.

These piezoelectric devices require the use of impedance matching layers to partially reduce the huge impedance mismatch between the air and the piezoelectric materials. In some cases, use of these layers is quite hard because they are either extremely light, brittle or thin. Therefore, such transducers have to be constructed using real materials that are far from the ideal specifications (Gómez 2004).

Taking into account these limitations, specific modelling and simulation tasks of different transducers design configurations, are needed. Afterward, these tasks can conduct to a more proper and complete transducer prototyping.

Therefore, the objective of this work is to develop and check a modelling tool (based on PSPICE) for the search of improved transducer configurations. This tool is verified by comparing the modelled transducers response with the experimentally measured one for the case, already known, of a design based in the use of two quarter-wavelength matching layers. In addition, the response of an ideal transducer (i.e. a transducer having two matching layers whose impedances are exactly those theoretically calculated) is also modelled. This provides a reference to compare the performance of transducers produced using real materials.

2. TRANSDUCER DESIGN

A wideband air-coupled piezoelectric transducer can be represented by a multi-element structure. Figure 1 presents the diagram of an air-coupled transducer design with two matching layers, no backing and mounted on aluminium housing. This comprises of a piezoelectric ceramic (the active element of the structure where the electro-mechanic energy conversion takes place), and the matching layers, which allow a more efficient transmission of the ultrasonic wave to the propagation medium. Additionally, other elements can be considered in the design depending of the requirements demanded by the transducer specific application, (for example, one element added to the piezoelectric ceramic rear face and named “backing”).

The enormous impedance mismatch between piezoceramics (≈30 MRayl) and air (≈0.0004 MRayl) has two important consequences for the design of air-coupled transducers: sensivity is very low (~ -60 dB) and bandwidth is quite narrow (~ 5%). Sensivity can be improved by a single matching layer, but widening the frequency bandwidth requires the use two or more matching layers. The selection of the suitable matching layer materials is a key issue.

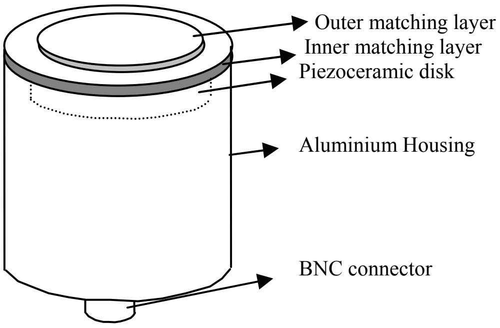


Figure 1. Diagram of the two-matching layers air-coupled transducer design.

Aditionally the thickness of the layers is a critical factor and several matching configurations have been proposed (Kelly, Hayward, and Gómez 2004, Gómez 2004, Toda 2002). In any of these matching configurations, a key aspect for the successful design is the acoustic impedance of outer layer. This is highly limited by the availability of suitable materials having the very low specific acoustic impedance, very low attenuation, and the required thickness for the designed configuration and working frequency.

The optimum acoustic impedance of the quarter wavelength layer can be obtained by the following expression:

$$Z_l = \sqrt{Z_1 Z_2} \tag{1}$$

where Z_l is acoustic impedance of the matching layer. Z_1 and Z_2 are the acoustic impedance of the piezoceramic and the air respectively.

For a stack of matching layers, the equation (1) can be generalized and the impedance of successive layers to be added can be also obtained.

An analysis of a group of the more proper materials for the construction of the matching layers at 1 MHz was done. Table I summarizes the properties required by the simulation model of some candidate materials to produce quarter wavelength matching layers to match piezoelectric transducers to the air using a two-layers scheme. In Table I *Z* is the specific acoustic impedance, f_r is the quarter wavelength resonant frequency and α is the attenuation coefficient. The polyurethane, the epoxy resin and the silicone rubber listed in Table I can be used as the first or inner matching layer, while the membranes in Table I can be used as the second or outer matching layer. Membrane materials, unlike the other available materials, are only available at certain grades and thicknesses, hence, for a given membrane material and grade f_r is fixed and can not be changed. The only way to obtain a different f_r is to change the grade, the material or the vendor (Gómez 2003a and 2003b)

The properties of the selected materials were analyzed and considered for the realization of distinct transducer design alternatives, including one or two matching layers. This analysis was complemented with modelling and simulation tasks of different transducer design options evaluated inside a selected practical transceiver configuration.

Table I. Materials properties

Material	Z (MRayl)	f_r (MHz)	α at f_r (Np/m)
Polyurethane	2.25	1.00	45
Epoxy resin	2.80	1.00	40
Silicone Rubber	1.35	1.00	45
Nylon membrane (1)	0.11	1.00 (*)	270
Polyethersulfone membrane (1,2)	0.25	1.20 (*)	200
Cellulose Nitrate (1,2)	0.25	1.00 (*)	400
Cellulose Nitrate (3)	0.12	0.7 (*)	680

1: Gómez (2003a)
2: Gómez (2004)
3: Kelly, Hayward, and Gómez 2004.
*: Frequency is given thickness and membrane grade. They can not be changed to tune the resonant frequency.

3. ULTRASONIC TRANSCEIVER MODEL

A through-transmission (T-T) configuration was considered in order to simulate the performance of the different design alternatives. This transceiver configuration was implemented in a circuit analysis program (PSPICE). Figure 2 presents the electric diagram of the pulsed T-T arrangement, used in the simulation process.

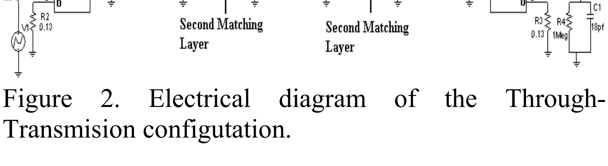


Figure 2. Electrical diagram of the Through-Transmision configutation.

The piezoelectric stages were modeled using a PSPICE implementation, of a well-known equivalent circuit [Redwood 1961], and which included a quadratic frequency dependence of the mecanical losses in the piezoceramic material [Ramos A., Ruíz A., San Emeterio J.

L., Sanz P.T., 2006]. The three port blocks (PZ27), symbolize the emitting-receiving probes, constructed from piezoceramic PZ27 (Ferroperm piezoceramics a/s). Pins e, b and f denote respectively the electrical and mechanical (back and front) ports of the transducers. Additionally, the matching layers were modelled employing lossy transmission lines using the corresponding material acoustical properties as input data [Van Deventer J., Torbjörn L., Jerker D., 2000].

The electronic driving stage was modelled by a pulsed source and a resistor, and considering the practical driving conditions employed. The receiving electronic included, some elements which symbolize the input impedance of the oscilloscope used. Figure 3 presents the driving electrical signal provided by the pulse source for the experimental measurements. This was generated by a Panametrics 5077 pulser/receiver. This is one half negative cycle of a square wave. The width of the pulse is tuned to provide a maximum energy output at the transducer centre frequency (1 MHz in this case). A similar signal was used for modeling purposes.

For all cases it was considered that the transmitter transducer was very similar to the receiver transducer and that they were separated by an air gap 5 cm thick.

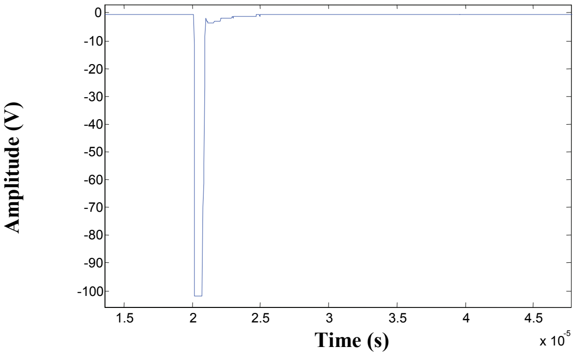


Figure 3. Electrical signal provided by the pulse source.

4. RESULTS.

4.1. Modelling of the ideal transducer response.

The performance of distinct design alternatives is analyzed by means of the transceiver characteristic broadband functions, such as through-transmission response in time and frequency domain. Among the different design options evaluated, we first considered the employment of matching layers with ideal materials, in order to achieve a reference. This reference was used to evaluate the results obtained with realistic configurations using real materials that present attenuation and whose impedance values do not exactly coincide with the theoretically required value.

This ideal case consists of a piezoceramic layer made of PZ27, with thickness resonant frequency located at 1 MHz. We also considered that matching layers were perfectly tuned to this frequency. For the case of one ideal and lossless quarter-wavelength matching layer the acoustic impedance (according to Eq. 1) takes the value of 0.11 MRayls. For the case with two ideal and lossless quarter-wavelength matching layers, the first or inner matching layer has an acoustic impedance of 0.73 MRayl while the second or outer matching layer has an acoustic impedance of 0.0017 MRayl. It is worthwhile to consider the fact that while for the one matching layer configuration it is possible to find some solid materials having an impedance close to the theoretically required value (at the cost of a non-negligible attenuation), there are not any known solid material with such a low acoustic impedance value as required by the outer matching layer in the two layers configuration.

Figure 4 presents the comparison of the simulated Through-Transmission (T-T) Frequency Response considering different design alternatives in the frontal face of the air-coupled transducer. The solid curve represents the (T-T) response without matching layer in the frontal face of the transducer and the dashed and dotted curves, with one and two matching layers, respectively. The ripple observed at low frequencies is produced by the signal reverberations produced within the air gap. A notable increment of the amplitude can be observed for both cases were a matching layer was used, respect to the case without matching layer. Additionally, a significant bandwidth increment is observed for the two-matching layer option.

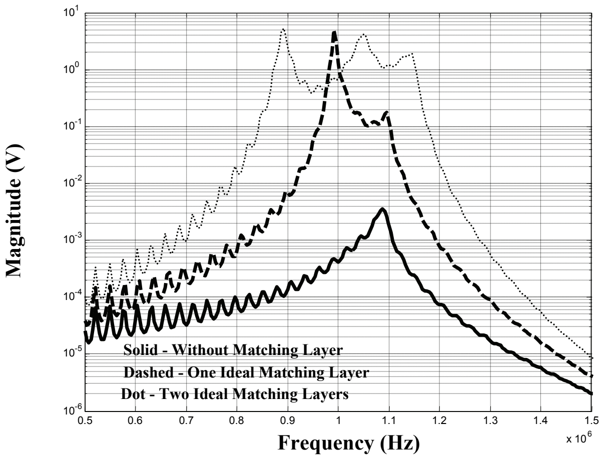


Figure 4. Through-Transmission Frequency Response considering different design alternatives in the frontal face of the air-coupled transducer. Solid: Transducer without matching layer. Dashed: Transducer with an ideal matching layer. Dotted: Transducer with two ideal matching layers.

4.2. Modelling of the best possible transducer using real materials and comparison with a practical realisation.

As a second step we consider the problem of transducer design using real materials. A list of some of the candidate materials was presented in Table I. The more successful results in the distinct simulation options evaluated were obtained with the polyurethane and the nylon membrane. According to this, two identical prototype transducers made of PZ27 piezoceramic and two quarter-wavelength matching layers (polyurethane and nylon membrane) were built. The transducer response is both measured and modelled at three different stages: 1. before attaching any matching layer, 2. after the attachment of the first (or inner) one but before attaching the second (or outer) one, 3. after both matching layers have been attached. This is useful for the control of the fabrication process, because in this way the integrity and correct functioning of each matching layer can be determined.

In Figure 5 the Simulated Through-Transmission Temporal Response considering different design alternatives in the frontal face of the air-coupled transducer, for the selected optimum design, employing real and low attenuating materials, is presented. In this Figure we can clearly appreciate the through transmitted signal from the transmitter to the receiver (this is the signal that arrives first) and also the first reverberation within the air gap. As the distance between transducers is 5 cm, this reverberation appears about 300 μs later than the through transmitted one. A bigger amplitude in the (T-T) waveforms is observed with the addition of one and two matching layers in the frontal face of the piezoceramic.

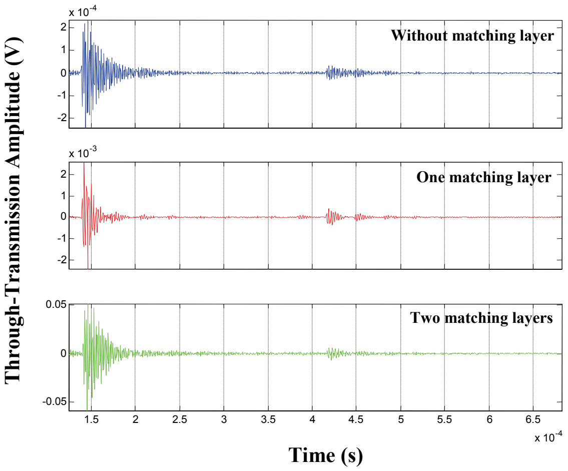


Figure 5. Simulated Through-Transmission Temporal Response considering different design alternatives in the front face of the air-coupled transducer, for a selected optimum design employing real and low attenuating materials. Blue: Transducer without matching layer. Red: Transducer with a matching layer. Green: Transducer with two matching layers.

Then a pair of transducers following this specifications (i.e. PZ27 piezoceramic and two quarter-wavelength matching layers, the first made of polyurethane and the second of nylon membrane) were built. Figure 6 shows a picture of one of them.

Figure 7. shows the experimentally measured Through-Transmission Temporal Response. In this case, separation between transmitter and receiver was 2 cm. The transmitter transducer was driven by the electrical pulse shown in Figure 3, the electrical signal generated at the receiver transducer were digitized and stored using a digital oscilloscope (TDS 5052). As observed in the simulation, the pulse amplitude increases notably with the addition of matching layers. In this case, the first reverberation within the air gap between transducers is also observed, though only about 120 μs later than the through transmitted signal, because the air gap length is smaller (2 cm).

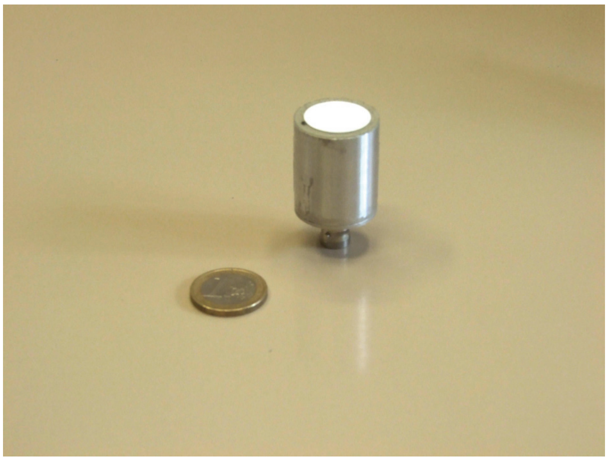


Figure 6. Picture of the air-coupled piezoceramic transducer constructed according to the proposed design i.e. with two matching layers in the frontal face).

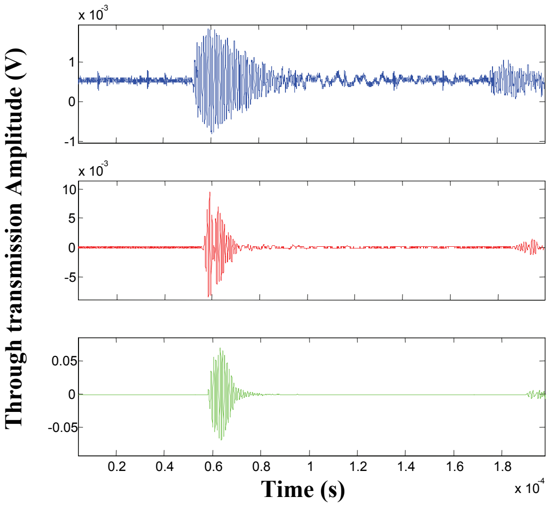


Figure 7. Measured Through-Transmission Temporal Response for several transducers configurations. 2cm separation air gap. Blue: Transducer without matching layer. Red: Transducer with a matching layer. Green: Transducer with two matching layers.

Simulated and experimental Through-Transmission (T-T) Frequency Responses (for the cases presented in Figure 5), can be observed in Figure 8. The theoretically calculated response shows the low frequency ripple mentioned before due to the signal reverberations within the air gap. This is not observed in the experimental data because a rectangular temporal window was applied to the signal to filter out these reverberations before the FFT is calculated. It is possible to appreciate bigger amplitude and bandwidth in the design alternatives with one and two matching layers respectively with respect the no matching layer option. Although the one layer option increases the transducer performance, it is further improved by the addition of a second matching layer. In general, the resonances experimentally observed are not so sharp as those theoretically predicted. This is due to possible imperfections of the layers (i.e. lack of homogeneity or flatness) and also to the possible contribution of the layer of glue used to attach the nylon membrane to the polyurethane layer.

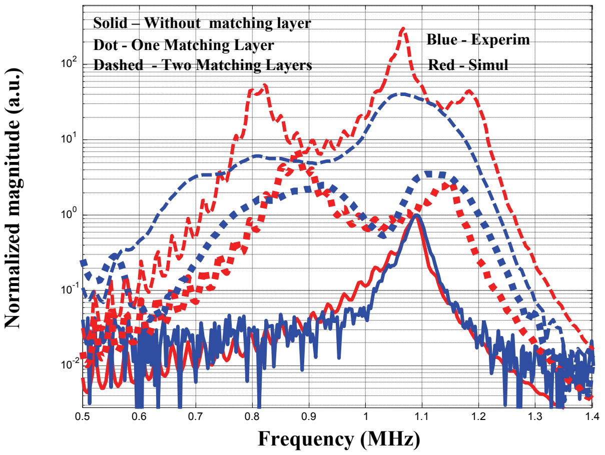


Figure 8. Comparison of the simulated (red color) and experimental (blue color) curves of the Through-Transmission Frequency Response, for different design alternatives in the frontal face of the air-coupled transducer. Solid: Transducer without matching layer. Dashed: Transducer with and ideal matching layer. Dotted: Transducer with two ideal matching layers.

4.3. Modelling of the influence of the impedance of the backing material.

The temporal pulse length emitted by the transducer is influenced by the characteristics of the backing element. The backing is usually a a highly attenuative, high density material that is used to control the vibration of the transducer by absorbing the energy radiating from the back face of the piezoceramic. This pulse length is closely related with the bandwidth as the shorter the pulse duration (or length) is, the larger the bandwidth of the transducer. Therefore a way to enlarge the frequency bandwidth of a transducer is to introduce a backing or to employ a material having larger acoustic impedance. This bandwidth enlargement is performed at the cost of reducing the sensitivity of the transducer. As sensitivity is a key issue in air-coupled transducers the suitability of this approach will be determined by the exact trade off between bandwidth enlargement and sensitivity decrease. To study this effect, the model developed and tested in this paper is especially useful.

In this case, we study the selected optimum design, employing real and low attenuating materials for the matching layers in the front face of the piezoceramic (see previous sections) together with three different backing options: no backing (original solution), 2 MRayl backing, and 7 MRayl backing. The second alternative (2 MRayl) corresponds to a light backing material, this can be achieved by using a silicon rubber. The third option (7 MRayl) corresponds to a heavy backing that can be achieved by using an epoxy resin loaded with tungsten powder.

Results are shown in Figure 9. This Figure presents a comparison among the Simulated Through-Transmission Temporal Responses considering the three different design alternatives in the rear face of the air-coupled transducer mentioned above. A shorter pulse length in the (T-T) waveforms can be observed with the addition of a backing, in the rear face of the piezoceramic. As mentioned above this pulse reduction is concomitant with a bandwidth enlargement. A clear reduction of the sensitivity (signal amplitude) is also observed. However in order to determine the best optimum backing impedance it is necessary to consider the application the transducer is intended for. In some cases (i.e. inspection of highly attenuating or difficult to penetrate materials) transducers sensitivity is the main design criteria, while for other cases (spectral analysis– Gómez 2003b and 2007-, study of dispersion relations – Gómez, de la Fuente and González-Gomez 2006 and Gómez and González-Gomez 2007-, axial resolution, search of thickness resonances, Gómez 2003a and 2003b,), the key issue is the bandwidth of the transducers.

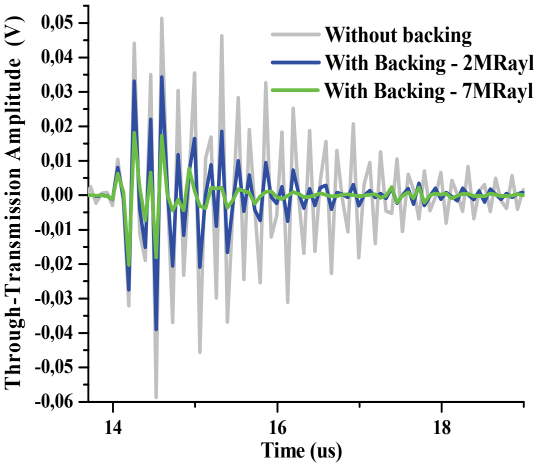


Figure 9. Simulated Through-Transmission Temporal Response considering different design alternatives in the rear face of the air-coupled transducer, for a selected optimum design employing real and low attenuating materials for the frontal matching layers. Black: Transducer without matching layer. Blue: Transducer with a backing of 2 MRayl. Green: Transducer with a backing of 7 MRayl.

5. CONCLUSIONS

A new approach to model air-coupled piezoelectric transducers with improved bandwidth and only considering real and low attenuating materials, is presented.

This model has been used to determine the best configuration for a 1 MHz air-coupled transducer based on a PZ27 piezoceramic and only considering real materials to produce matching layers. Configurations based on one and two quarter-wavelength matching layers has been studied. The result of this study is a transducer configuration based on a double matching layer made of polyurethane and nylon membrane. In addition, the performance of ideal air-coupled transducers using ideal matching layers has also been modelled. This provides a reference value in order to compare the performance of real transducers. Finally, the utility of the use of different backing materials has been modelled in order to determine the possibility to further increase the bandwidth of the proposed solution.

According to the model predictions, a pair of prototype air-coupled transducers was built. The model predictions have been compared with the experimentally measured performance of the prototype transducers. This provides an initial verification of the model capability.

REFERENCES

Van Deventer J., Torbjörn L., Jerker D., 2000. Pspice Simulation of Ultrasonic Systems, *IEEE Trans. Ultrason. Ferroelect. Freq. Contr.,* 47 (4), 1014-1024.

Grandia, W. A. and Fortunko, C. M., 1995, NDE applications of air-coupled ultrasonic transducers. *Proc. IEEE Ultrason. Symp.,* pp. 697-708.

Fortunko, C. M., Renken, M. C., and Murray, A., 1990, Examination of objects made of wood using air-coupled ultrasound. *Proc. IEEE. Ultrason. Symp.,* 1099-1103.

Gómez Álvarez-Arenas, T. E., Montero, F., Moner-Girona, M. Rodríguez, E. Roig, A. Molins, E. 2002, Viscoelasticity of silica aerogels at ultrasonic frequencies. *Appl. Phys. Lett.* 81 (7), 1198-1200.

Gómez Álvarez-Arenas, T. E., 2003a. Air-coupled ultrasonic spectroscopy for the study of membrane filters, *J. Membr Sci*. 213, 195-207.

Gómez Álvarez-Arenas, T. E., 2003b. A Nondestructive integrity test for membrane filters based on air-coupled ultrasonic spectroscopy, *IEEE Trans. Ultrason. Ferroelect. Freq. Contr.,* 50 (6), 676-685.

Gómez Álvarez-Arenas, T. E., 2004. Acoustic impedance matching of piezoelectric transducers to the air, *IEEE Trans. Ultrason. Ferroelect. Freq. Contr.,* 51 (5), 624-633.

Gómez Álvarez-Arenas, T. E., de la Fuente, S., González-Gómez, I. 2006, Simultaneous determination of apparent tortuosity and microstructure length scale ane shape: Application to rigid open cell foams. *Appl. Phys. Lett.* 88, 221910.

Gómez Álvarez-Arenas, T. E. González-Gómez, I. 2007, Spatial normalization of the high-frequency ultrasound energy loss in open-cell foams. *Appl. Phys. Lett*., 90, 201903.

Gómez Álvarez-Arenas, T. E., Apel, P. Yu. And Orelovitch, O. L., 2007. Characterization of ion-track membranes by non-contact ultrasonic magnitude and phase spectroscopy. *J. Membr. Sci.,* 301, 210-220.

Hutchins, D. A. and Schindel, 1994. D. W., Advances in non-contact and air-coupled transducers. *Proc. IEEE Ultrason Symp.*, 1245-1254.

Kelly, S.P., Hayward, G., and Gómez Álvarez-Arenas, T. E., 2004. Characterization and assessment of an integrated matching layer for air-coupled ultrasonic applications, *IEEE Trans. Ultrason., Ferroelect., Freq. Contr.,* 51 (10) 1314-1323.

McIntyre, C. S., Hutchins, D. A., Billso, D. R. and Stor-Pellinen, J., 2001. The use of air-coupled ultrasound to test paper. *IEEE Trans. Ultrason., Ferroelect., Freq., Contr*., 48 (3), 717-727.

Redwood, M. 1961, Transient Performance of a Piezoelectric Transducer. J. Acoustic. Soc. Amer., 33 (4), 527-536.

Ramos A., Ruíz A., San Emeterio J. L., Sanz P.T., 2006. Pspice circuital modelling of ultrasonic imaging transceivers including frequency-dependent acoustic losses and signal distortions in electronic stages. Ultrasonics, 44, e995–e1000.

Toda, M. 2002, New type of matching layers for air-coupled ultrasonic transducers. IEEE Trans. Ultrason., Ferroelect., Freq., Contr., 49 (7), 972-979.

WORKPLACES EFFECTIVE ERGONOMIC DESIGN: A LITERATURE REVIEW

Antonio Cimino[a], Duilio Curcio[b], Francesco Longo[c], Giovanni Mirabelli[d]

[a] [b] [c] [d] Modeling & Simulation Center - Laboratory of Enterprise Solutions (MSC-LES)
Mechanical Department, University of Calabria, 87036 Rende (CS), Italy

[a]acimino@unical.it, [b]dcurcio@unical.it, [c]f.longo@unical.it, [d]g.mirabelli@unical.it

ABSTRACT

The paper proposes a literature review on the workplaces ergonomic effective design in the manufacturing systems and industrial plants sector. The main objective is to provide the reader with an accurate overview on the main scientific approaches proposed (during the last decades) by researchers and scientists working in this specific area. The paper passes through the description of several research works as they run through the literature. The initial search identifies a huge number of articles which were reduced to about 50 studies based on content and quality. The descriptive analysis of the literature reveals heterogeneity in the content of the scientific approaches due to the different principles, methods and tools applied for improving the interaction between humans and their working environment.

Keywords: manufacturing systems, ergonomics, effective ergonomic design, workplaces, workstations.

1. INTRODUCTION

The high complexity of manufacturing systems in terms of interaction between humans and their working environment continuously provides challenging problems for researchers working in this specific field.

An ergonomic approach to the design of an industrial workplace (ergonomic effective design) attempts to achieve an appropriate balance between the worker capabilities and worker requirements, to optimize worker productivity, as well as provide worker physical and mental well-being, job satisfaction and safety.

During the last years this research area has became more and more important due to its effects on system efficiency and productivity. In this regards, different research works have been proposed and several scientific approaches have been developed trying to achieve the ergonomic effective design of the workplaces belonging to the manufacturing system.

It is the intent of the paper to present a literature review on this specific area clustering the high quality research works according to the scientific approach they propose. In this regards, the authors identify three different scientific approaches based on different principles, methods and tools. The description of the research works for each scientific approach represents the core part of this literature review.

Before getting into the details of the study, in the sequel a brief overview of each section of the paper is presented. Section 2 describes the scientific approach based on the use of video tape systems. Section 3 presents a number of research works using several ergonomic standards for achieving the ergonomic effective design. Section 4 discusses about the third scientific approach based on the interaction between ergonomics and work measurement aspects. Section 5 presents briefly an ergonomic effective design application example based on a scientific approach proposed by the authors.

Finally, the last section reports the conclusions that summarize the scientific contribution of the work.

2. VIDEO TAPE SYSTEMS FOR THE ERGONOMIC WORKPLACES DESIGN

The evaluation of the ergonomic risk levels affecting a workplace represents the first step for achieving the ergonomic effective design. In industrial plants, for existing workplaces the ergonomic risks can be assessed through observation (Karhu et al. 1981). In this context, a video tape based approach is easy and time saving (Vedder and Hellweg, 1998). In effect the interference of video camera with the tasks being performed by the observed worker is minimal. However, note that if the operations require to move to different plant areas, multiple cameras have to be used. Neverthless, during the years a number of research works proposes the use of the video tape systems as main tool for the ergonomic effective design. Such research works are here presented as they run through the literature.

Hagström et al. (1985) and Engström et al. (1987) use video recording respectively in the meat-cutting and vehicle design research areas.

Das and Sengupta (1996) provide the guidelines for a good workstation design by observing workstation procedures and collecting data by video taping the operators as well (an application example is proposed in the field of supermarket checkstand workstations).

Engstrom and Medbo (1997) develop a video based observation method for time data collection and analysis of work time consumption. The method allows to measure the efficiency of the production system by separating between value-added and not value-added works activities.

Vedder (1998) presents an easy-to-use video-based posture analysis method for workplaces, where tasks interference have to be minimized and postures have to be observed over a longer period of time. The author identifies hazardous postures and their causative factors and then decides the appropriate re-design measures.

The approach based on video tape systems for data collection and analysis has been also used by Kadefors et al. (2000). In this case the video film is displayed on the computer terminal for evaluating (by using an interactive procedure) workers' ergonomic problems (pain and discomfort).

Neumann et al. (2001) present a video-based posture assessment method capable of measuring trunk angles and angular velocities in industrial workplaces.

Forsman et al. (2002) propose a method based on video recordings synchronized with physiological measurements for characterizing work time consumption and physical work load of manual work. The method was developed throught two casies studies within the Swedish automotive industry. It is concluded industrial interventions could be designed by means of such method.

Actually the use of the video tape could generate a vast amount of recordings which are tedious to analyze. Even in this case, such scientifc approach allows to identify the tasks causing hazardous postures and suggest appropriate redesign measures as well. In this regards, Vedder and Hellweg (1998) record twenty days and nights shifts in a fibbre spinning area of a chemical plant by means of a stationary camera. A very long analysis of the videotapes allows them to provide the guidelines for a correct redesign of the system under consideration.

3. ERGONOMIC STANDARDS

The second scientific approach regards the application of ergonomic standards as support tools for the ergonomic effective design. Among the ergonomic standards, the following have to be regarded as the most widely used: the NISOH 81 and the NIOSH 91 equations for lifting tasks (NIOSH stands for *National Institute for Occupational Safety and Health*); the OWAS analysis for analyzing working postures (OWAS stands for *Ovako Working Analysis System*); the RULA method for estimating the risks of work-related upper limb disorders (RULA stands for *Rapid Upper Limb Assessment*);

In the sequel research works are introduced according to the ergonomic standard used. The section consists of 5 subsections. Three subsections for presenting the research works concerning the most widely used ergonomic standards (one subsection for each ergonomic standard). The forth subsection is then reported for introducing the less used ergonomic standards: the OCRA methods for analyzing worker's exposure to tasks featuring various upper-limb injury risk factors (OCRA stands for *Occupational Repetitive Action*); the Garg analysis for assessing the energy expenditure for performing an operation; the Burandt-Schultetus analysis for lifting tasks involving a large number of muscles. In conclusion, the last subsection proposes the research works based on the integration of several ergonomic standards.

Before getting into the details of each subsection, a brief description of the ergonomic standard under consideration is provided.

3.1. NIOSH 81 and NIOSH 91 method

NIOSH 81 and NIOSH 91 evaluate the ergonomic risk levels affecting the lifting tasks.

The NIOSH 81 method calculates the action limit (AL) and the maximum permissible limit (MPL). AL is the weight value which is permissible for 75% of all female and 99% of all male workers. MPL is the weight value which is permissible for only 1% of all female and 25% of all male workers.

The NIOSH 91 analysis, additionally to the NIOSH 81, includes the recommended weight limit (RWL) and the lifting index (LI). The RWL is the load that nearly all healthy workers can perform over a substantial period of time for a specific set of task conditions. The LI is calculated as ratio between the real object weight and the Recommended Weight Limit.

Further information about the cited ergonomic standards can be found in Niosh Technical Report (1981) and Waters et al. (1994).

Let us present the research works aiming at achieving the workplace ergonomic effective design by means of NIOSH analysis.

Grant et al. (1995) analyze musculoskeletal trauma among preschool workers in the United States by means of NIOSH methods. The authors evaluate the possible causes of back and lower extremity pain among 22 workers at a Montessori day care facility. Finally they present recommendations for modifying the workplace and changing the organization and methods of work as well.

Grant et al. (1997) evaluate the magnitude of lifting hazards in the shipping department of a wooden cabinet manufacturing company. The representative lifts are analyzed using the Revised National Institute for Occupational Safety and Health (NIOSH) Lifting Equation. The results suggest that work in shipping department imposes a high level of physical demand, which may increase the risk of work related back pain and other musculoskeletal injury. In this regards the authors provide recommendations for reducing physical workload through automation, introduction of mechanical assists, changes in work organization and more frequent job rotation.

Mital and Ramakrishnan (1999) analyze a complex manual materials handling task, which involved lifting, turning, carrying, and pushing activities, by using both the old and revised NIOSH lifting guidelines (Niosh Technical Report 1981; Waters et al., 1993) as well as the guidelines provided by Mital et al. (1993, 1997).

Hermans et al. (1999) evaluate the effect of using a mechanical device on physical load during the end assembly of cars. According to the NIOSH equation, 8 out of 10 of the tasks should only be performed by trained workers and preferably with tools.

Chung and Kee (2000) analyze lifting tasks using the 1991 revised NIOSH lifting equations for a fire brick manufacturing company with a high prevalence of low back injuries. The results suggest that the tasks should be redesigned ergonomically to eliminate the risk factors that may cause low back injuries. The authors propose a tasks redesign based on making horizontal locations closer to a worker or reducing the symmetric angles.

Temple and Adams (2000) use the NIOSH analysis in order to establish ergonomic acceptable limits for an industrial lifting station. Through the analysis of several factors the authors define a cumulative lifting index and use such index for detecting ergonomic problems during lifting tasks. They successively modify the lifting station for reducing ergonomic risks and preventing lower back related injuries.

Lin and Chan (2007) carry out an ergonomic workstation re-design for reducing musculoskeletal risk factors and musculoskeletal symptoms among female's workers of a semiconductor fabrication room. By means of walk-through observations of the working environment, discussing with company's managers and using NIOSH analysis, the authors identify the most prevalent and urgent ergonomic issues to be resolved and modify the layout of the workplace for reducing ergonomic hazards.

3.2. OWAS Analysis

The OWAS analysis carries out a qualitative analysis of the worker's movements during a working process. The analysis calculates the stress associated to each body posture and classifies them in one of the following four stress categories:

- Category 1: the stress level is optimum, no corrective interventions are required;
- Category 2: the stress level is almost acceptable, corrective interventions are necessary in the near future;
- Category 3: the stress level is high, corrective interventions are required as soon as possible;
- Category 4; the stress level is very high, corrective interventions must be carried out immediately.

Further information about the cited ergonomic standard can be found in Kharu et al. (1981).

During the last years several research works have adopted the OWAS analysis for evaluating the workers body postures.

Carrasco et al. (1995) describe an ergonomic evaluation of three different designs of checkouts workstation, which require the operators to stand when they scan the products, pack them into plastic bugs and transfer the packed bags to the customer. Musculoskeletal load and exertion associated with the different checkouts are measured using the OWAS analysis. The results of the evaluation form the basis of recommendations for an improved workstation design.

Nevala-Puranen et al. (1996) analyze physical workload and strain when milking in a parlor. OWAS analysis is accomplished for evaluating the postural load. The authors assert that the information of this study can be utilized in the development of the working environment of milking.

Scott and Lambe (1996) implement the OWAS in a poultry industry. The authors apply the ergonomic analysis highlighting wrong postures and providing the guidelines for an improved workstation design.

Van Wendel de Joode et al. (1997) conduct a workplace survey in order to quantify the physical load in a population of male workers in two ships maintainace companies. The Ovako Working Postire Analizyng System is used for measuring the postural load. The results reveal that awkward postures of the back occurr in 38% of the work time and the stress on the neck/shoulder region due to one or both arms above shoulder level is present in 25% of the work time.

White and Kirby (2003) use the OWAS analysis for evaluationg health-care workers in the methods used to fold and unfold selected manual wheelchairs. The authors conclude that many of the methods used include bent and twisted back postures that are known to be associated with a high risk of injury.

Perkiö-Makelä and Hentilä (2005) estimate the physical workload and strain of dairy farming in loose housing barns. The feeding and removing manure and spreading of bedding are analyzed by means of OWAS analysis. On the basis of the analysis results, the authors provide some recommendations for the building of new loose-housing barns (for example, providing enough space for automated feeding and cleaning systems).

3.3. RULA method

RULA is a postural targeting method for estimating the risks of work-related upper limb disorders. A RULA assessment gives a quick and systematic assessment of the postural risks to a worker. The analysis can be conducted before and after an intervention to demonstrate that the intervention has worked to lower the risk of injury. The RULA action levels give you the urgency about the need to change how a person is working as a function of the degree of injury risk.

- Action level 1: it means the person is working in the best posture with no risk of injury from their work posture;
- Action level 2: it means that the person is working in a posture that could present some risk of injury from their work posture, so this should be investigated and corrected;
- Action level 3: it means that the person is working in a poor posture with a risk of injury from their work posture, and the reasons for

this need to be investigated and changed in the near future to prevent an injury;

- Action level 4: it means that the person is working in the worst posture with an immediate risk of injury from their work posture, and the reasons for this need to be investigated and changed immediately to prevent an injury.

A full description of the RULA method is contained in McAtamney and Corlett (1993).

In the last decades, several authors have used the RULA method as support tool for achieving the workplace ergonomic effective design.

Gonzáles et al. (2003) evaluate the relationship between the ergonomic design of workplaces and achieved product quality levels. In particular, a metalworking firm with ISO-9002 certification was selected, and its quality results were analyzed with respect to reprocessed and rejected parts after varying the initial work method on the basis of the results of an ergonomic evaluation by means of RULA. It was concluded that a reduction in ergonomic problems implies better quality records.

Massaccesi et al. (2003) investigate work related disorders in truck drivers using the RULA method. A sample of 77 drivers, of rubbish-collection vehicles who sit in a standard posture and of roadwashing vehicles, who drive with the neck and trunk flexed, bent and twisted, is studied. After the analysis, the authors conclude that ergonomic interventions aiming at modifying the truck's workstation are recommended for preventing musculo-skeletal disorders.

Choobineh et al. (2004) propose ergonomic intervention in carpet mending operations. Seventy-two menders are questioned regarding musculoskeletal disorders. Based on the problems found, a new workstation is developed and eight menders are asked to work in the new workstation. They are observed and evaluated with the RULA technique and their opinion on the improvement is asked working on four frequently seen tasks. The new workstation improves working posture noticeably.

Shuval and Donchin (2005) propose an application of the RULA method in the HI-TECH industry. Results of the RULA underline the need for implementing an intervention program focusing on arm/wrist posture.

3.4. Others ergonomic standards

Here the OCRA method, the Garg and Burandt Schultetus analysis are briefly described.

3.4.1. OCRA methods

The Occupational Repetitive Action methods (OCRA) analyze worker's exposure to tasks featuring various upper limb injury risk factors (repetitiveness, force, awkward postures and movements, lack of recovery periods). The OCRA methods are the OCRA index and the OCRA checklist. The OCRA index can be predictive of the risk of upper extremity work related musculoskeletal disorders in exposed populations. It is generally used for the (re)-design or in depth analysis of workstations and tasks (Colombini et al. 1998, 2002). The OCRA checklist, based on the OCRA index, is simpler to apply and is generally recommended for the initial screening of workstations featuring repetitive tasks (Occhipinti et al. 2000; Colombini et al. 2002).

The OCRA method is based on a consensus document of the International Ergonomics Association (IEA) technical committee on musculoskeletal disorders (Colombini et al. 2001). Further information regarding OCRA methods can be found in Occhipinti and Colombini (1996).

3.4.2. Burandt Schultetus analysis

The Burandt-Schultetus analysis allows evaluating the load limits for a specific working posture (keeping into consideration the weight of the grasped objects). The Burandt-Schultetus analysis is usually applied to lifting activities in which a large number of muscle groups are involved. The main result is the maximum weight (Permissible Limit, PL) that the worker can lift. The Permissible Limit can be evaluated by using equation (1):

$$PL = G * C * AJ * RF \qquad (1)$$

- G is a coefficient for the worker's gender;
- C is a coefficient for the worker's health condition;
- AJ is a coefficient for worker's age and type of job;
- RF is the reference force.

Note that the AJ (Age and Job factor) depends on the effort type (i.e. static or dynamic), the worker's age, the shift time (i.e. 8 hours) and the effort frequency. The RF takes into consideration the torso weight movement, the hands use (i.e. one or two hands), the number of persons performing the operation (i.e. one or two persons), the effect of secondary jobs and the maximum force. In turns, the torso weight movement depends on the lower and upper grasp height and motion frequency; the maximum force depends on body size class (anthropometric measure), upper grasp height and distance of grasp from the body.

The maximum permissible force is then compared to the current actual force (AF) being exerted. Three different cases can be distinguished:

- Case 1: the maximum permissible force does not exceed the actual force then an ergonomic intervention is required;
- Case 2: the maximum permissible force is equal to the actual force, then a corrective intervention is necessary in the near future;
- Case 3: the actual force is lower than the maximum permissible force, then no ergonomic intervention is required.

Further information can be found in Schultetus (1980).

3.4.3. Garg analysis

The Garg analysis calculates the total amount of energy spent during the manual operations. The analysis splits up a specified operation into smaller steps calculating for each of them the Energy Expenditure (EE); the sum of these separate steps represents the total Energy Expenditure for the activity. As input parameters, such analysis requires information concerning load weight and body weight as well as gender of the working person. Further information can be found in Garg (1976).

3.5. Ergonomic standards integration

In order to achieve relevant ergonomic improvements some authors propose an effective ergonomic design based on the integration of different ergonomic standards.

Wright and Haslam (1999) investigate manual handling risks within a soft drinks distribution centre using the OWAS postural analysis and the NIOSH equations. The authors compare two working methods involving pallets and cages. The analysis detects significant manual handling risks and reports musculoskeletal disorders.

Jones et al. (2005) present an examination of three common pub occupations (bartending, waitressing and cooking). Risk of musculoskeletal injury is evaluated for the three occupations analyzed by means of RULA method and NIOSH Lifting Equation. Finally recommendations for reducing the risks are provided.

Jones and Kumar (2007) quantify physical exposure information collected from 15 saw-filers in four sawmill facilities by means of the RULA, REBA, ACGIH TLV, Strain Index and OCRA procedures based on multiple posture and exertion variable definitions.

Russell et al. (2007) compare the results of different ergonomic standards (NIOSH, ACGIH TLV, Snook, 3DSSPP and WA L&I) for evaluating ergonomic risks in lifting operations. Each ergonomic standard is applied to a uniform task (lifting and lowering two different types of cases) with the aim of choosing the best work methods by appropriately interpreting the results of the ergonomic analysis.

4. ERGONOMICS AND WORK MEASUREMENT

Another important issue to take into consideration in the workplace design is the strict relation between the concepts of work measurement and ergonomics. The measurement of the work aims at evaluating the time standard for performing a particular operation. On the contrary, the concept of ergonomics is often indicated as study of work (Zandin 2001) and studies the principles that rule the interaction between humans and their working environment. In effect, the work measurement and the ergonomics affect each other: ergonomics changes affect the time required for performing the operations as well as any change to the work method affects the ergonomics of the workplace.

Different research works have taken into consideration both ergonomics and work measurement aspects.

Das and Sengupta (1996) propose a workstation design procedure based on the optimization of the worker and total system productivity as well as worker physical and mental well-being, job satisfaction and safety.

Resnick and Zanotti (1997) underline that ergonomic principles can potentially be used to improve productivity as well. An application example is proposed for remarking that a workstation can be designed to maximize performance and reduce costs by considering both ergonomics and productivity together.

Laring et al. (2002) develop an ergonomic complement to a modern MTM system called SAM. In particular the authors propose a tool that gives the possibility to estimate simultaneously the consumption of time in the envisaged production and the biomechanical load inherent in the planned tasks.

Udosen (2006) propose a tools for construction, evaluation and improvement (in terms of ergonomic and time issues) of a workplace for the assembly of a domestic fan.

Another important issue cited in many research works developed in the last decades of the 20th century is the application of the ergonomic standards and work measurement methods directly in the real system.

Usually such approach requires a huge amount of money and time for exploring all the possibilities in terms of workstations configurations, work assignment, works methods, etc. Therefore researchers and practitioners started to develop research works by using Modelling & Simulation (M&S) as support tool for choosing correctly, for understanding why, for diagnose problems and explore possibilities (Banks, 1998). From an animation point of view, the simulation provides virtual three-dimensional environments that strongly support the workstation ergonomic design. A three-dimensional visualization is certainly an important support that can be used to detect problems and critical factors that otherwise would be difficult to detect.

Wilson (1997) proposes an overview on attributes and capabilities of virtual environments (devoted to support ergonomic design) and describes a framework for their specification, development and evaluation.

Marcos et al. (2006) aim at reducing the stress of the medical staff during laparoscopic operations simultaneously increasing the safety and efficiency of an integrated operation room. To this end, the authors develop a simulator by integrating the CAD software (CATIA) and the simulation software (RAMSIS).

Over the years the M&S approach has became more and more appealing thanks to the numerous advantages such as the possibility to study ergonomic issues at the earliest stages of design in order to avoid potential future ergonomic redesign in the real-world system.

Feyen et al. (2000) propose a PC-based software program (based on the integration of a Three-Dimensional Static Strength Prediction Program, 3DSSPP, for biomechanical analysis with a widely used computer-aided design software package, AutoCAD). As consequence, the authors are able to study ergonomic issues during the design phase taking into consideration different design alternatives.

Chang and Wang (2007) propose a method for conducting workplace ergonomic evaluations and re-design in a digital environment with the aim of preventing work-related musculoskeletal disorders during assembly tasks in the automotive sector.

Longo et al. (2006) use M&S in combination with ergonomic standards and work measurement for the effective design of an assembly line still not in existence. The authors propose a multi-measures approach with the aim of obtaining a different work assignment to each workstation, better line-balancing and better ergonomic solutions.

Santos et al. (2007) propose an ergonomic study on working positions in a manufacturing company (by using the simulation software eM-Workplace) and providing, as result, remarkable ergonomic improvements. In particular, the study is based on the integration of several ergonomic standards (NIOSH 81, NIOSH 91, Burandt Schultetus, OWAS and Garg analysis) and the Method Time Measurements (MTM) methodology.

5. APPLICATION EXAMPLE

The authors propose their scientific approach for the ergonomic effective design by means of a real case study. The case study regards the most critical workstation (the Seal Press workstation) of a manufacturing process devoted to produce high-pressure hydraulic hoses. The effective ergonomic design of the workstation takes into consideration both ergonomic risks and work measurement. The actual workstation configuration is compared with several alternative scenarios by using a well planned experimental design. To this end, the authors propose an approach based on multiple design parameters and multiple performance measures with the aim of considering both the interaction of the operators with their working environment and the work methods. In addition, the authors use Modelling & Simulation (M&S) as a support tool for implementing a three-dimensional environment capable of recreating, with satisfactory accuracy, the real Seal Press Workstation.

5.1. Simulation model development

The first step was the development of a simulation model capable of recreating the production process of the workstation. The simulation model development involves three different phases: collecting data concerning the Assembly area (data collection phase), reproducing the real system in the virtual environment from both a geometric and work method point of view (simulation modelling phase) and verifying if the simulation model is an accurate representation of the real system (validation phase).

Figure 1 shows a panoramic view of the virtual layout of the Seal Press Workstation.

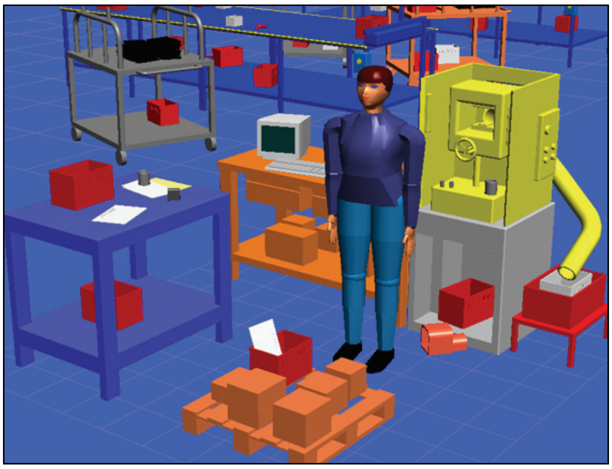


Figure 1: Simulation model of the Seal Press workstation

5.2. Design of Experiment

A well-planned Design of Experiments (DOE) is used for supporting the comparison of the actual configuration of the Seal Press workstation with alternative operative scenarios (different workstation configurations). The DOE requires to select a set of design parameters (a group of factors to be changed during the simulation runs). We take into consideration the following factors:

- *Support table angle*: let us indicate this angle with α, it defines the orientation of the support table respect to the actual position (see figure 2);
- *Raw materials bin height*: let us indicate this height with *rmh*, it defines the height of the bin containing the raw materials (see figure 2);
- *Ring nuts bin height*: let us indicate this height with *rnh*, it defines the height of the bin containing ring nuts exiting from the seal press machine (see figure 2).

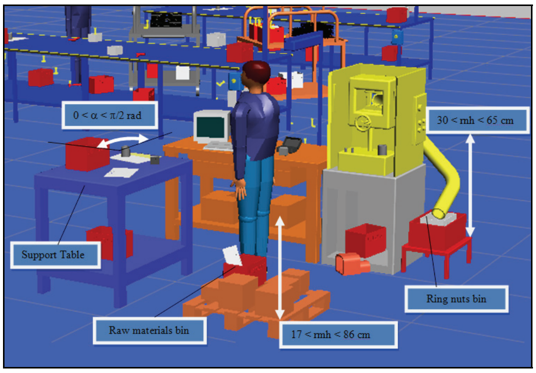


Figure 2: Actual configuration of the Seal Press workstation (with design parameters).

Note that the figure 2 shows the actual configuration of the Seal Press workstation.

Table 1 reports factors and levels; the factors levels combinations create a comprehensive set of different scenarios in terms of workstation layout and tools disposition (8 different configurations to be tested with the simulation model).

Table 1: Design parameters and levels

Seal Press Workstation				
Factors	**Factor ID**	**Level 1**	**Level 2**	
Support Table Angle	α	0	π/2	rad
Raw Materials bin height	rmh	17	86	cm
Rings nuts bin height	rnh	30	65	cm

As previously stated, the effective ergonomic design of a workstation consider a multi-measures approach based on ergonomic and work measurement indexes.

The ergonomic performance measures, based on ergonomic standards, are the lift index (evaluated by using the Burandt-Schultetus analysis), the stress level associated to each working posture (evaluated by using the OWAS analysis) and the energy expenditure associated to each activity (evaluated by using the Garg analysis). The most important performance measure for work measurement is the process time; we use the Method and Time Measurement methodology (MTM-1) for evaluating the process time.

5.3. Simulation results and workstation final configuration

The experiments before described (8 different configurations to be tested with the simulation model) have been completely carried out by using the simulation model, monitoring for each alternative scenario the multiple performance measures. Table 3 reports the simulation results.

The authors analyze the effects of each design parameter on the performance measures and according to such analysis develop a new workstation configuration.

Figure 3 shows the effective ergonomic re-design of the Seal Press workstation (final design).

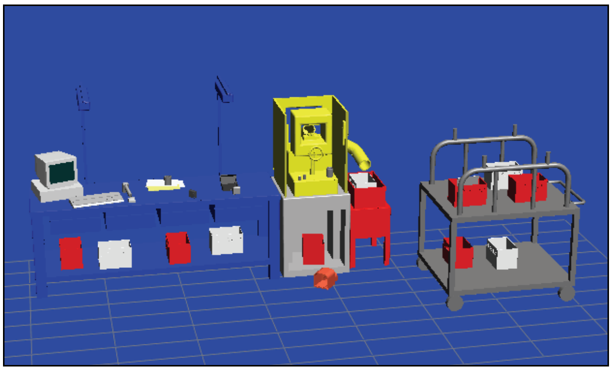


Figure 3: Effective ergonomic design of the Seal Press workstation.

Further research works on workstation ergonomics effective design using Modeling & Simulation combined with ergonomic standards and work measurement can be found in Longo et al. (2005), Longo et al. (2006-a), Longo et al. (2006-b), De Sensi et al. (2007-a), De Sensi et al. (2007-b), Bocca and Longo (2008).

6. CONCLUSIONS

The main objective of the paper is to present a literature review concerning the ergonomic effective design. The initial search identifies a huge number of articles which were reduced to about 50 studies based on content and quality. The research works were clustered according to the scientific approach they propose. In this regards, the authors identify three different scientific approaches based on different principles, methods and tools.

Several authors propose an approach based on the use of video tape systems for evaluating the ergonomic risks affecting the workplaces. Note that such evaluation represents the first step for achieving an ergonomic effective design.

Table 3: Simulation results

Seal Press Workstation							
			Burandt-Schultetus		**OWAS**	**Garg**	**MTM-1**
α	**rmh**	**rnh**	**Permissible Force (N)**	**Actual Force (N)**	**Stress Level**	**Energy Expenditure (Kcal)**	**Process Time (sec)**
0	17	30	121.3	147.2	3	1480.0	470.32
0	17	65	135.0	147.2	2	1438.8	464.75
0	86	30	137.7	147.2	2	1403.6	460.23
0	86	65	151.4	147.2	1	1362.4	454.66
π/2	17	30	121.3	147.2	3	1439.4	456.71
π/2	17	65	135.0	147.2	2	1398.3	451.14
π/2	86	30	137.7	147.2	2	1363.0	446.62
π/2	86	65	151.4	147.2	1	1321.9	441.05

A number of research works propose the application of ergonomic standards. The review identifies NISOH 81, NIOSH 91, OWAS and RULA as the most widely used ergonomic standards.

The third scientific approach regards the interaction between ergonomics and work measurement aspects. In this regards, the authors identify two different thought tendencies: (i) the application of ergonomic standards and work measurement methods directly in the real system; (ii) the application of ergonomic standards and work measurement methods by means of Modelling & Simulation (M&S) as support tool for the ergonomic effective design.

Finally, the literature review is completed with a scientific approach proposed by the authors for achieving the ergonomic effective design of workplaces. Note that such scientific approach is explained by means of an application example.

REFERENCES

Banks. J., 1998. *Principles of Simulation, Handbook of Simulation*. New York: Wiley Interscience.

Bocca, E., Longo, F., 2008. *Simulation Tools, Ergonomics Principles and Work Measurement Techniques for Workstations Design*. Proceedings of Summer Computer Simulation Conference, 15-18 June, Edinburgh (UK).

Carrasco, C., Coleman, N., Healey, S., 1995. Packing products for customers, An ergonomics evaluation of three supermarkets checkouts. *Applied Ergonomics*, 26, 101-108.

Chang, S.-W., Wang, M.-J. J., 2007. Digital Human Modeling and Workplace Evaluation: Using an Automobile Assembly Task as an Example. *Human Factors and Ergonomics in Manufacturing,* 17, 445-445.

Choobineh, A., Tosian, R., Alhamdi, Z., Davarzanie, M., 2004. Ergonomic intervention in carpet mending operation. *Applied Ergonomics,* 35, 493-496.

Chung, M.K., Kee, D., 2000. Evaluation of lifting tasks frequently performed during fire brick manufacturing processes using NIOSH lifting equations. *International Journal of Industrial Ergonomics,* 25, 423-433.

Colombini, D., Grieco, A., Occhipinti, E., 1998. Occupational musculoskeletal disorders of the upper limbs due to mechanical overload. *Ergonomics,* 41, 1347-1356.

Colombini, D., Occhipinti, E., Delleman, N., Fallentin, N., Kilbom, A., Grieco, A., 2001. Exposure assessment od upper limb ripetitive movements: a consensus document, in *International Encyclopaedia of Ergonomics and Human Factors,* London: Taylor & Francis.

Colombini, D., Occhipinti, E., Grieco, A., 2002. *Risk Assessment and Management of Repetitive Movements and Exertions of Upper Limbs*, Vol. 2 New York: Elsevier.

Das, B., Sengupta, A. K., 1996. Industrial workstation design: a systematic ergonomics approach. *Applied Ergonomics*, 27, 157-163.

De Sensi, G., Longo, F., Mirabelli, G., 2007-a. *Modeling & Simulation Based Approach for Optimizing Seal Press Workstation in a Manufacturing System*, Proceedings of Business and Industry Symposium, March 25-29, USA.

De Sensi, G., Longo. F., Mirabelli, G., 2007-b. *Ergonomic work methods optimization in a three dimensional environment*, Proceedings of Summer Computer Simulation Conference, July 15-18, San Diego, California, USA

Engström, T., Medbo, P., 1997. Data collection and analysis of manual work using video recording and personal computer techniques. *International Journal of Industrial Ergonomics,* 19, 291-298.

Feyen, R., Liu, Y., Cha, D., Jimmerson, G., Joseph, B., 2000. Computer-aided ergonomics: a case study of incorporating ergonomics analyses into workplace design. *Applied Ergonomics,* 31, 291-300.

Forsman, M., Hansson, G.Å., Medbod, L., Asterland, P., Engström, T., 2002. A method for evaluation of manual work using synchronised video recordings and physiological measurements. *Applied Ergonomics*, 33, 533-540.

Garg A., 1976. *A metabolic rate prediction for manual materials handling jobs*. Dissertation, University of Michigan.

Gonzáles, B.A., Adenso-Dìaz, B., Torre, P.G., 2003. Ergonomic performance and quality relationship: an empirical evidence case. *International Journal of Industrial Ergonomics,* 31, 33-40.

Grant, K.A., Habes, D.J., Bertsche, P.K., 1997. Lifting hazards at a cabinet manufacturing company: Evaluation and recommended controls. *Applied Occupational and Environmental Hygiene*, 12, 253-258.

Grant, K.A., Habes, D.J., Tepper, A.L., 1995. Work activities and musculoskeletal complaints among preschool workers. *Applied Ergonomics,* 26, 405-410.

Hagström, P., Engström, T., Magnusson, M., Örtengren, R., 1985. Ergonomics in meat-cutting II, Investigation of physical work-load and efficiency in different systems for materials handling. *Proceedings of Ninth Congress of the International Ergonomics Association*, pp. 250-252. Bournemouth.

Hermans, C., Hautekiet, M., Spaepen, A., Cobbaut, L., De Clerq, J., 1999. Influence of material handling devices on the physical load during the end assembly of cars. *International Journal of Industrial Ergonomics,* 24, 657-664.

Jones, T., Kumar, S., 2007. Comparison of ergonomic risk assessments in a repetitive high-risk sawmill occupation: Saw-filer. *International Journal of Industrial Ergonomics,* 37, 744-753.

Jones, T., Strickfaden, M., Kumar, S., 2005. Physical demands analysis of occupational tasks in neighborhood pubs. *Applied Ergonomics,* 36, 535-545.

Kadefors, R., Forsman, M., 2000. Ergonomic evaluation of complex work: a participative approach employing video computer interaction, exemplified in a study of order picking. *International Journal of Industrial Ergonomics,* 25, 435-445.

Kharu, O., Harkonen, R., Sorvali, P., Vepsalainen, P., 1981. Observing working postures in industry: Examples of OWAS application. *Applied Ergonomics*, 12, 13-17.

Laring J., Forsman M., Kadefors R., Örtengren R., 2002. MTM-based ergonomic workload analysis. *International Journal of Industrial Ergonomics*, 30, 135-148.

Lin, R.T., Chan, C.-C., 2007. Effectiveness of workstation design on reducing musculoskeletal risk factors and symptoms among semiconductor fabrication room workers. *International Journal of Industrial Ergonomics,* 37, 35-42.

Longo, F., Mirabelli, G., Papoff, E., 2005. *Techiche di analisi avanzate per la progettazione efficiente delle postazioni di assemblaggio manuale*, SdA – Soluzioni di Assemblaggio, VNU Business Publications Italia.

Longo, F., Mirabelli, G., Papoff, E., 2006-a. *Material Flow Analysis and Plant Lay-Out Optimization of a Manufacturing System*, International Journal of Computing, 5(1), 107-116.

Longo, F., Mirabelli, G., Papoff, E., 2006. Effective Design of an Assembly Line Using Modeling & Simulation. *Proceedings of the Winter Simulation Conference*, Monterey, California, USA.

Marcos, P., Seitz, T., Bubb, H., Wichert, A., Feussner, H., 2006. Computer simulation for ergonomic improvements in laparoscopic surgery. *Applied Ergonomics,* 37, 251-258.

Massaccesi, M., Pagnotta, A., Soccetti, A., Masali, M., Masiero, C., Greco, F., 2003. Investigation of work-related disorders in truck drivers using RULA method. *Applied Ergonomics,* 34, 303-307.

McAtamney, L., Corlett, E.N., 1993. RULA: a survey method for the investigation of work-related upper limb disorders. *Applied Ergonomics*, 24, 91-99.

Mital, A., Nicholson, A.S., Ayoub, M.M., 1993. *A Guide to Manual Materials Handling*. London: Taylor & Francis.

Mital, A., Nicholson, A.S., Ayoub, M.M., 1997. *A Guide to Manual Materials Handling*. 2nd ed. London: Taylor & Francis.

Mital, A., Ramakrishnan, A., 1999. A comparison of literature-based design recommendations and experimental capability data for a complex manual materials handling activity. *International Journal of Industrial Ergonomics,* 24, 73-80.

Neumann, W.P., Wells, R.P., Norman, R.W., Kerr, M.S., Frank, J., Shannon, H.S., OUBPS Working Group, 2001. Trunk posture: reliability, accuracy, and risk estimates for low back pain from a video based assessment method. *International Journal of Industrial Ergonomics,* 28, 355-365.

Nevala-Puranen, N., Kallionpää, M., Ojanen, K., 1996. Physical load and strain in parlor milking. *International Journal of Industrial Ergonomics,* 18, 277-282.

Niosh Technical Report 81-122. National Institute for Occupational Safety and Health (Hrsg.). Work practices guide for manual lifting. *Center for Disease Control, U.S. Department of health and human services*, Cincinnati, OH, USA: NTIS 1981.

Occhipinti, E., Colombini, D., 1996. Alterazioni muscolo-scheletriche degli arti superiori da sovraccarico biomeccanico: metodi e criteri per l'inquadramento dell'esposizione lavorativa. *Medicina del Lavoro,* 87, 491-525.

Occhipinti, E., Colombini, D., Cairoli, S., Baracco, A., 2000. Proposta e validazione preliminare di una checklist per la stima dell'esposizione lavorativa a movimenti e sforzi ripetuti degli arti superiori. *Medicina del Lavoro*, 91, 470-485.

Perkiö-Makelä, M., Hentilä, H., 2005. Physical workstrain of dairy farming in loose housing barns. *International Journal of Industrial Ergonomics,*35, 57-65.

Resnick, M.L., Zanotti, A., 1997. Using ergonomics to target productivity improvements. *Computer & Industrial Engineering,* 33, 185-188.

Russell S. J., Winnemuller L., Camp J. E., Johnson P. W., 2007. Comparing the results of five lifting analysis tools. *Applied Ergonomics*, 38, 91-97.

Santos, J., Sarriegi, J. M., Serrano, N., Torres, J. M., 2007. Using ergonomic software in non-repetitive manufacturing processes: A case study. *International Journal of Industrial Ergonomics,* 37, 267-275.

Schultetus, W., 1980. Daten, hinweise und beispiele zur ergonomischen arbeitsgestaltung. Montagegestaltung, Verlag TÜV Rheinland Gmbh, Köln.

Scott, G. B., Lambe, N. R., 1996. Working practices in a pherchery system, using the OVAKO Working posture analysing System (OWAS). *Appied Ergonomics,* 27, 281-284.

Shuval, K., Donchin, M., 2005. Prevalence of upper extremity musculoskeletal symptoms and ergonomic risk factors at a Hi-Tech company in Israel. *International Journal of Industrial Ergonomics,*35, 569-581.

Temple, R., Adams, T., 2000. Ergonomic Analysis of a Multi-Task Industrial Lifting Station Using the NIOSH Method. *The Journal of Industrial Technology*, 16, 1-6.

Udosen U.J., 2006. Ergonomic workplace construction, evaluation and improvement by CADWORK. *International Journal of Industrial Ergonomics,* 36, 219-228.

Van Wendel de Joode, B., Burdorf, A., Verspuy, C., 1997. Physical load in ship maintenance: Hazard evaluation by means of a workplace survey. *Applied Ergonomics,* 28, 213-219.

Vedder, J., 1998. Identifying postural hazards with a video-based occurrence sampling method. *International Journal of Industrial Ergonomics*, 22, 373-380.

Vedder, J., Hellweg, 1998. Identifying postural hazards with a video-based occurrence sampling method. *International Journal of Industrial Ergonomics*, 22, 4-5.

Waters, T.R., Putz-Anderson, V., Garg, A., Fine, L.J., 1993. Revised NIOSH equation for the design and evaluation of manual lifting tasks. *Ergonomics*, 36, 749-776.

Waters, T.R., Vern, P.A., Garg, A., 1994. Application Manuals for the Revised NIOSH Lifting Equation. *U.S. Department of health and human services, National Institute for Occupational Safety and Health*, Cincinnati, OH, USA.

White, H.A., Kirby, R.L., 2003. Folding and unfolding manual wheelchairs: an ergonomic evaluation of health-care workers. *Applied Ergonomics,* 34, 571-579.

Wilson J.R., 1997. Virtual environments and ergonomics: needs and opportunities. *Ergonomics,* 40, 1057-1077.

Wright E.J., Haslam R.A., 1999. Manual handling risks and controls in a soft drinks distribution centre. *Applied Ergonomics*, 30 , 311-318.

Zandin K.B., 2001. *Maynard's Industrial Engineering Handbook*, 5th ed. New York: McGraw-Hill.

AUTHORS BIOGRAPHY

ANTONIO CIMINO was born in Catanzaro (Italy) in October the 1st, 1983. He took his degree in Management Engineering, summa cum Laude, in September 2007 from the University of Calabria. He is currently PhD student at the Mechanical Department of University of Calabria. His research activities concern the integration of ergonomic standards, work measurement techniques, artificial intelligence techniques and Modeling & Simulation tools for the effective workplace design. He collaborates with the Industrial Engineering Section of the University of Calabria to research projects for supporting innovation technology in SMEs.

DUILIO CURCIO was born in Vibo Valentia (Italy), on December the 15th, 1981. He took the degree in Mechanical Engineering from University of Calabria (2006). He is currently PhD student at the Mechanical Department of University of Calabria. His research activities include Modeling & Simulation and Inventory Management theory for production systems and Supply Chain design and management. He collaborates with the Industrial Engineering Section of the University of Calabria to research projects for supporting Research and Development in SMEs.

FRANCESCO LONGO took the degree in Mechanical Engineering from University of Calabria (2002) and the PhD in Industrial Engineering (2005). He is currently researcher at the Mechanical Department (Industrial Engineering Section) of University of Calabria. His research interests regard modeling & simulation of manufacturing systems and supply chain management, vulnerability and resilience, DOE, ANOVA. He is Responsible of the Modeling & Simulation Center – Laboratory of Enterprise Solutions (MSC-LES), member organization of the MS&Net (McLeod Modeling & Simulation Network) He is also member of the Society for Computer Simulation International and Liophant Simulation.

GIOVANNI MIRABELLI was born in Rende in 1963 and he took the degree in Industrial Engineering at the University of Calabria. He is currently researcher at the Mechanical Department of University of Calabria. His research interests include ergonomics, methods and time measurement in manufacturing systems, production systems maintenance and reliability, quality. He has published several scientific papers participating as speaker to international and national conferences. He is actively involved in different research projects with Italian and foreign universities as well as with Italian small and medium enterprises.

AUTHOR’S INDEX

Abdul Wahab, 231
Affenzeller, 59, 69, 75, 95, 100, 106
Aguilar, 481, 499, 693
Aguirre, 159
Álvarez Castelló, 204
Ambrogio, 140
Anthony, 342
Apsley, 610
Arkatkar, 770
Atri, 392
Babarada, 312, 333
Bahroun, 730
Bai, 841
Balaguer, 261
Balderud, 231
Baquero, 499
Barrientos, 738
Baruwa, 579
Beham, 75, 100, 106
Beit, 744
Benna, 353
Beraldi, 655
Bicocchi, 665
Biswas, 841
Bögl, 33
Bondarciuc A., 312
Bondarciuc V., 312
Booth, 610
Bordegoni, 140
Bossomaier, 506
Bouché, 212
Braune, 27
Breitenecker, 386, 392, 398, 408, 419, 425
Bruni, 655
Bruno F., 134, 140
Bruno P., 718
Bruzzone, 521
Burt, 610
Calì, 780
Campagne, 730
Can, 87
Capel Tuñón, 274
Capra, 493
Careddu, 243
Carothers, 284
Caruso F., 134, 140
Caruso G., 140
Castilla Rodríguez, 481, 693
Caussanel, 438
Cerezo, 218
Cesarotti, 224
Chaczko, 7, 15, 20, 45
Champliaud, 146
Chapurlat, 706
Chiu, 20
Chourabi, 675
Cicirelli, 447, 455
Cimino, 620, 754, 788, 817, 853
Clarke, 534, 627
Copilau, 323
Corona-Armenta, 431
Corzo, 204
Costa, 472
Costantino, 243
Costea, 348
Cuesta, 218
Curcio, 620, 754, 824, 853
Cutolo, 647
D'acquisto, 487
Dalessandro, 841
Dalle, 515
Dammasch, 368
Dao, 146
De Almeida, 683
De Back, 300
De Nicola, 647

De Pinho, 472
De Prada, 154, 188, 806
Del Cerro, 738
Del Re, 69
Demongodin, 255
Di Gravio, 243
Di Gregorio, 665
Di Salle, 718
Di Silvio, 224
Dias, 688
Diep, 706
Dotoli, 700
Dubitzky, 300
Dudenhoeffer, 267
Dunare, 333
Dvornik J., 724
Dvornik S., 724
Edl, 198
Eitzinger, 386
Ekiz, 537
El Mashade, 556
El Tokhy, 556
Elders, 610
Emrich, 408
Evangelisti, 665
Ever, 602
Fanti, 700
Fedunov, 306
Feng, 146
Ferretti, 180
Ferrise, 140
Francisco, 169
Franz, 1
Frattaruolo, 638
Furfaro, 447, 455
Cunha, 683
Gaion, 712
Galloway, 610
Garcia, 261
García-Hevia, 481, 499
Gattamelata, 112
Gemikonakli, 602
Gharbi, 744
Giambiasi, 831
Giordano, 447
Gómez Álvarez-Arenas, 847
Gonzalez M., 665
González Santos, 204
Gradisar, 374
Grau, 218
Gronalt, 353
Guerriero, 655
Gullì, 569
Gulyas, 300
Gunal, 693
Gutierrez, 738
Gyimesi, 386, 425
Heavey, 87
Heidl, 386
Hennet, 255
Hildebrandt, 594
Hirsch, 69
Hofmann, 586
Holder, 284
Horton, 368, 632
Ingrassia, 126
Introna, 224
Jacak, 39
Jaganathan, 534
Jolly, 237
Jovan, 374
Kaczorek, 362
Kampis, 300
Katebi, 231
Kaup, 237, 627
Kenne, 744
Kincaid, 534
Klempous, 45
Koçyiğit, 602
Kofler, 100

Kolditz, 594
Korecký, 198
Krieger, 586
Kronberger, 75, 95, 106
Krull, 632
Kurowski, 300
Leal, 472
Legato, 561, 569
Li, 431
Llarrull, 831
Longo, 527, 546, 817, 824, 853
Lopes, 683
López, 218
Löscher, 419
Lupu, 316, 323
Lütjen, 174
Mahmoud, 556
Malone, 627
Manea, 333
Mangini, 700
Manic, 267
Manolescu, 323
Manzo, 638, 647
Marinov, 348
Marins, 472
Martinez, 738
Massei, 521
Mayr, 1
Mazaeda, 188
Mazza, 561
Medina-Marín, 431
Mendes, 683
Méndez, 159
Meneghini, 718
Mercantini, 255
Merkuryeva, 51
Mininel, 712, 718
Mirabelli, 620, 754, 788, 817, 853
Mlak, 333
Moalla, 730
Moldovan, 333
Montaño-Arango, 431
Montevechi, 472
Montúfar-Benítez, 431
Mori, 224
Mroue, 438
Muguercia Torres, 119
Mujica, 800
Müller, 159
Muñoz, 481
Murphy, 87
Music, 380
Musmanno, 134
Muzzupappa, 134
Nagele, 392
Naldi, 487
Napalkova, 51
Narciso, 800
Niculiu, 312, 323, 333
Nigrelli, 126
Nigro, 447, 455
Nikodem, 45
Noga, 806
Norman, 610
Novak, 1
Oleson II, 237, 627
Oliveira, 688
Oliveri, 780
Ören, 527, 546
Ortiz, 841
Palacin, 154
Papoff, 620, 754, 817
Pereira, 688
Pérez-Lancho, 169
Pérez-Rojas, 431
Pezzuti, 112
Pidd, 693
Piera, 579, 800, 811
Pinto, 655
Pisacane, 134

Pollet, 675
Popa, 358
Popper, 398, 408, 425
Posset, 353
Pröll, 39
Pupo, 455
Qing, 841
Ramos Fernández, 342
Rarità, 638
Ravariu C., 312, 333
Ravariu F., 312
Rebai, 706
Rende, 134
Ribault, 515
Rinaldi, 463
Rosa, 627
Rossainz López, 274
Rossi, 738
Ruíz Toledo, 342, 847
Rusu, 312, 333
Saetta, 762
San Martin, 738
Sanei, 602
Sanz, 169
Sarjoughian, 537
Sarri, 463
Scavotti, 521
Scholz-Reiter, 174, 594
Schuddebeurs, 610
Seffino, 159
Sequenzia, 780
Shehata, 15
Standish, 506
Sukov, 81
Suter, 339
Syafiie, 154
Szemes, 300
Tadeo, 154
Tauböck, 392
Thamizh Arasan, 770
Tiacci, 762
Tireli, 724
Tlili, 730
Tosato, 249
Tremori, 521
Trkovský, 198
Trovato, 780
Trunfio, 569
Ukovich, 700, 712
Ulrych, 198
Umeton, 665
Valentini, 112
Vatta, 712, 718
Vega, 169
Venuto, 134
Vesely, 1
Vilanova, 261
Vizireanu, 333
Votava, 198
Wagner, 59, 69, 75, 95, 100, 106
Wassertheurer, 408
Wiedemann, 293
Winkler, 59, 69, 95, 106
Xu, 632
Zamarreño Cosme, 204
Zanni, 212
Zanoni, 180
Zauner, 398, 408, 425
Zavanella, 180
Zengin, 537
Zorzut, 374
Zuñiga, 811