

Contents

- Foreword
- Committees
- I3M09 Program

- Invited Talks
- Access to Papers by Track
- Author Index

WELCOME TO EMSS 2009

This 21th EMSS edition asserts the importance of maintaining a high quality International forum for the Simulation Community in which different modeling approaches can be discussed in a friendly but also rigorous ambient. The inherent complexity of present industrial, business, biological, medicine and health care system, together with the changing mentality in the management of those systems under new sustainable rules, provoke the necessity for a better understanding of the internal system's dynamic.

Sometimes, there is a well accepted misunderstanding regarding the term "system complexity" due to a real lack of knowledge on system behavior that appears as emergent dynamics fostered by subsystem interactions. Non justified delays, congestions and idleness are not any longer tolerated by efficient and competitive production and supply chain systems. Present industrial crisis pushed a cross-roads moment in which efficiency should be the driver and the goal of most re-engineering processes. Thus, new design process oriented tools will be essential to tackle efficiency from the strategic and operational point of view in order to address the long term production problems considering also short term aspects. 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; therefore today M&S (Modeling and Simulation) is becoming even more important considering the necessity to introduce new economic, environmental and social sustainability issues.

We are sure that EMSS2009 forum will foster deeper academic and technical discussions on promoting simulation techniques as a basis for new generation decision support tools.

This year, and for the sixth time, the European Modeling and Simulation Symposium (EMSS2009) has been organized in the International Mediterranean 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 Simulation Council of the SCS International: International Mediterranean and Latin America Council of Simulation (I_M_CS, www.i-m-cs.org).

The very high quality level achieved by this conference, it is confirmed by the fact that also in 2009, two special issues in International Journals have been arranged for publication based on their best papers:

- Industry and Supply Chain: Technical, Economic and Environmental Sustainability.
Guest Editor: Francesco Longo
- Modeling & Applied Simulation: Advanced Methodologies and Techniques.
Guest Editors: Agostino Bruzzone, and Iván Castilla.

We wish you a fruitful conference and a pleasant stay in Puerto de la Cruz, Tenerife.

EMSS Proceedings Editors,

Rosa M. Aguilar, ISAATC, Universidad de La Laguna, Spain
Agostino G. Bruzzone, MISS-DIPTM, University of Genoa, Italy
Claudia Frydman, LSIS, Domaine Universitaire de Saint-JeromeLSIS, France

Proceedings of the European Modeling and Simulation Symposium, EMSS 2009
Vol I - ISBN 978-84-692-5414-1

Francesco Longo, MSC-LES, University of Calabria, Italy
Khalid Mekouar, Higher School of Engineering in Applied Sciences - ESISA
Miquel Angel Piera, MISS - Spanish Center, Univ. Autònoma de Barcelona, Spain

I3M Organizers



Academic Sponsors



Sponsors



SIMULATION AND OPTIMIZATION OF DYNAMICAL SYSTEMS

Cesar de Prada

Dpt. of Systems Engineering and Automatic Control, University of Valladolid, Spain

prada@autom.uva.es

ABSTRACT

This paper provides an overview of some methods and tools available for dealing with MIDO problems using simulation environments. The paper focuses in the so called sequential approach, where optimization algorithms are combined with a dynamic simulator in order to compute cost functions and constraints. A novel parameterization is proposed that allows converting the MIDO problem in a NLP one, saving computation time. The paper also examines the problems associated to the evaluation of the gradients when discontinuities take place in the inputs or the model due to its hybrid character.

Keywords: Dynamic Optimization, Hybrid systems, sequential methods, gradient computation.

1. INTRODUCTION

Simulation methods and tools are increasingly used in industry and other sectors for different purposes. Many times the interest of developing a simulation is a better understanding of the process or to train operators in its operation, but quite often the model is built with the aim of supporting decision-making. This is the case of design problems, advanced control or operation optimization. In these problems, the value of some decision variables has to be chosen so that a certain cost function is minimized or maximized, while a set of equality and inequality constraints among the variables are respected at the same time. This corresponds to the format of a typical optimization problem and, when the model is a dynamical one, to a dynamic optimization that can be formulated as:

$$\begin{aligned} \min_p \quad & J(p) = \int_0^T L(x, u) dt \\ & F(\dot{x}, x, u) = 0 \\ & g(x, u) \leq 0 \end{aligned} \quad (1)$$

where J is the cost function to be minimized and $u(p)$ is the vector of manipulated variables parameterized by the decision vector p along the prediction horizon T . x is the state vector and F and g represent the model dynamics and the constraints respectively.

Two main general methods have emerged for dealing with (1): simultaneous methods based on discretization of J , F and g , that give rise to a static

problem similar to (2), and sequential methods based on the calculus of J by means of the integration of F using a dynamical simulator, as in Fig.1.

$$\begin{aligned} \min_{p, x_i} \quad & J(p, x_i) = \sum_i L(x_i, u_i(p)) \\ & \phi(x_i, u_i) = 0 \\ & g(x_i, u_i) \leq 0 \end{aligned} \quad (2)$$

Notice that, in both cases, an adequate control vector parameterization is required in order to convert the problem from an infinite number of decision variables into a finite one. Also, the aim is to transform the dynamical optimization problem into a static one that can be solved with an appropriate NLP solver.

The main advantage of the simultaneous methods is its conceptual simplicity and the fact that path constraints can be translated and taken into account directly. Nevertheless, it creates a large number of additional decision variables, the states on every discretization point. In addition, the selection of the discretization scheme and the associated integration interval guaranteeing a certain integration error is not easy, in particular when stiff models are considered.

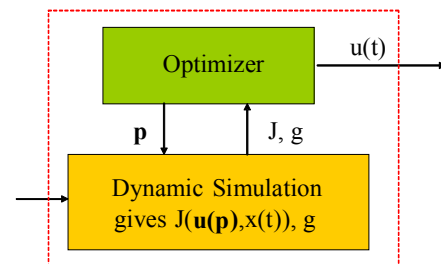


Figure 1

The basic idea of the sequential methods is depicted in Fig.1: Each time the optimizer requires the value of the cost function J (or the constraints violations g), it launches a dynamic simulation, passing it the value of the current p decision variables, which is used to compute J and g . The main advantage of this approach is the smaller number of decision variables (only p along the control horizon) as well as its structural simplicity. However, the path constraints are more difficult to enforce

Another point that is worth to mention is related to the numerical methods used to solve the NLP optimization problem. Most of them are based on the computation of gradients, so that calculating them correctly is crucial in the applicability of the methods. There are also alternatives that use stochastic methods like genetic algorithms, but our experience is that, at least if used alone, do not provide the required performance for real-time applications. Usually, gradients of J in relation to the decision variables p are not available directly and need to be computed, either by finite differences, or by the so called extended equations with provides the sensibilities $\partial x/\partial p$ that allow computing the gradient:

$$\frac{dJ}{dp} = \int_0^T \left(\frac{\partial L}{\partial x} \frac{\partial x}{\partial p} + \frac{\partial L}{\partial u} \frac{\partial u}{\partial p} \right) dt \quad (3)$$

The sensibilities can be obtained on-line computing the derivative of the model equations F with respect to p :

$$\frac{\partial F}{\partial \dot{x}} \frac{\partial \dot{x}}{\partial p} = \frac{\partial F}{\partial \dot{x}} \frac{d}{dt} \frac{\partial x}{\partial p} = \frac{\partial F}{\partial x} \frac{\partial x}{\partial p} + \frac{\partial F}{\partial u} \frac{\partial u}{\partial p} \quad (4)$$

Notice that denoting:

$$s = \frac{\partial x}{\partial p} \quad (5)$$

(4) can be written as:

$$\frac{\partial F}{\partial \dot{x}} \frac{ds}{dt} = \frac{\partial F}{\partial x} s + \frac{\partial F}{\partial u} \frac{\partial u}{\partial p} \quad (6)$$

which is a set of first order differential equations in s that can be integrated besides the model equations $F(\dot{x}, x, u) = 0$ to provide the sensibilities s and, by means of (3), the gradients.

One important remark is that the Jacobian of (6) and $F(\dot{x}, x, u)$ are the same, hence, the integration of the extended system only requires the Jacobian of the original model, saving time and providing an efficient way for computing gradients.

2. HYBRID SYSTEMS

In practice, it happens in many cases that the nature of the decision problem involves elements that can be of different nature or change the model over time. For instance, continuous decision variables that can be represented by real variables as well as other that can have only a finite number of states, like number of units in a design problem, or motors that operate only on/off, and are better represented by integer variables. In the same way, the operation can be described a set of logical rules, or the model can change its structure depending on its operating conditions. These processes involving continuous and discrete elements or variable

structure are known as hybrid systems and the associated optimization problems are denoting as mix integer dynamic optimization ones (MIDO). There are different modelling formalism for hybrid systems, but, from the point of view of the topic of this paper, they can be grouped in those that model the logic and discrete decisions using binary variables, (see Floudas 1995, or Bemporad et al. 2000) and those that use a finite automaton structure, more adequate to dynamic simulation. Considering the last ones, the dynamic optimization problem (1) can be formulated more specifically as:

$$\begin{aligned} \min_p \quad & J(p) = \int_0^T L(x, u) dt \\ & F^k(\dot{x}, x, u) = 0 \\ & g^k(x, u) \leq 0 \end{aligned} \quad (7)$$

where F^k represents each of the models associated to the transitions between different stages reflecting the discontinuities of the dynamic evolution of the process. The change between stage k and $k+1$ takes place when a transition function

$$\phi_k^{k+1}(x^k, u^k, t) \quad (8)$$

is activated, which typically occurs when it changes sign. At the transition time instant t_k , the system can jump its state to a new value x^{k+1} given by the so called transition condition T_k^{k+1} :

$$T_k^{k+1}(x^{k+1}, u^{k+1}, x^k, u^k) = 0 \quad (9)$$

See Fig.2 for an illustration, where t^* represents the time instant when the transition takes place.

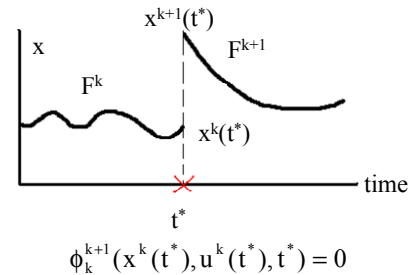


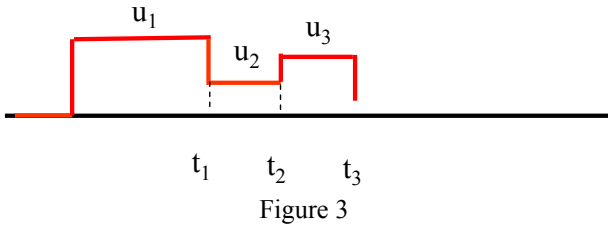
Figure 2

In the context of a sequential optimization methods, in the hybrid case, the simulator embeds the possible changing process dynamics which means that it has to support efficient algorithms for mixed continuous / discrete simulation, and, in particular a rigorous treatment of discontinuities. Notice that logic conditions and variable structure models are handled by the simulator and, in this sense, are hidden to the optimizer. Nevertheless, when applied to hybrid optimization problems, the computation of gradients presents problems associated to its special structure. See (Galan et al. 1999).

In general terms, two kinds of discontinuities can be distinguished: those associated to input parameterizations and those associated to transitions activated by the states. Let's examine how they affect the time evolution of the sensitivities s .

2.1. Input parameterization discontinuities

There are many ways of approximating an infinite valued signal $u(t)$ into a set of finite values. A typical way is represented in Fig.3:



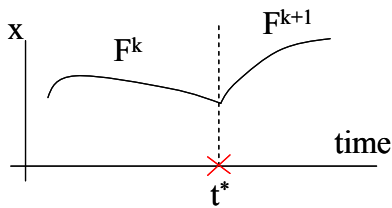
where the signal $u(t)$ takes the constant value u_i in the time interval $[t_{i-1}, t_i]$, with t_i representing the discontinuity time instants, so that:

$$\phi_k^{k+1} = t - t_k \quad (10)$$

Hence, the vector of decision variables can be represented by $p = [\dots, u_i, \dots]$ or by $p = [\dots, u_i, \dots, t_i, \dots]$ if the time instants t_i are considered as additional decision variables. Notice that, if the input is a binary variable, the related u_i s will be also binary ones.

As no state jump is associated to the input changes at $t_k = t^*$, the transition condition is given by:

$$T_k^{k+1} : x^{k+1}(t^*) - x^k(t^*) = 0 \quad (11)$$



When jumping from stage k to $k+1$ at t^* , see Fig.4, the new sensitivity equations (6) are given by:

$$\frac{\partial F^{k+1}}{\partial \dot{x}^{k+1}} \frac{ds_i^{k+1}}{dt} = \frac{\partial F^{k+1}}{\partial x^{k+1}} s_i^{k+1} + \frac{\partial F^{k+1}}{\partial u^{k+1}} \frac{\partial u^{k+1}}{\partial p_i} \quad (12)$$

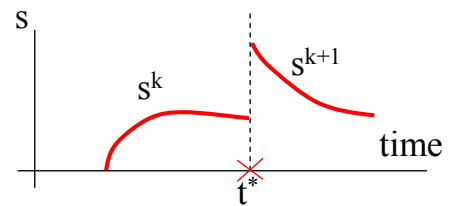
for each of the parameters p_i . As $s_i = \partial x / \partial p_i$, if the initial state is not a decision variable, the initial conditions for solving (6) at $t = 0$, will be $s_i = 0$. But, when considering (12), the initial conditions are related to the previous evolution of s^k and the transition condition (11). Computing the derivative of $T_k^{k+1}(x^{k+1}, x^k, t^*)$ with respect to p_i one gets:

$$s_i^{k+1}(t_k) = s_i^k(t_k) - [\dot{x}^{k+1}(t_k) - \dot{x}^k(t_k)] \frac{\partial t^*}{\partial p_i} \quad (13)$$

which is a key equation for what follows.

Two different cases can be distinguished now from the point of view of the time evolution of s_i : The first one is when $p = [\dots, u_i, \dots]$, so that the transition times are predefined values, typically $t_k = k\Delta$. In this case t^* does not depend on p_i and the sensitivities do not present any discontinuity. The second case includes "elastic" shape parameterizations in which some t_k are decision variables, $p = [\dots, u_i, \dots, t_i, \dots]$, so that, for those $p_i = t_i$, then $s_i^{k+1}(t^*)$ presents a jump, given by (13), (14) and directly related to the change at t_k of the time derivative of x , and depicted in Fig.5:

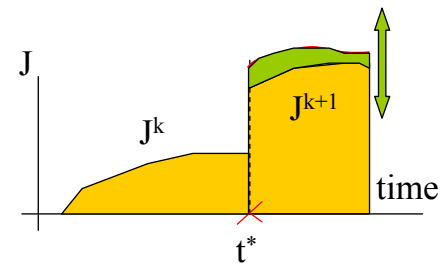
$$\begin{cases} p_i = u_i & \frac{\partial t}{\partial p_i} = 0 & s_i^{k+1}(t_k) = s_i^k(t_k) \\ p_i = t_k & \frac{\partial t}{\partial p_i} = 1 & s_i^{k+1}(t_k) = s_i^k(t_k) - [\dot{x}^{k+1}(t_k) - \dot{x}^k(t_k)] \end{cases} \quad (14)$$



Notice, nevertheless, that the fact that s_i presents a discontinuity in a time instant t_k does not imply that the gradient of J (related to the surface under the curve according to (3)) is discontinuous:

$$\frac{dJ}{dp_i} = \int_0^T \left(\frac{\partial L}{\partial x} s_i + \frac{\partial L}{\partial u} \frac{\partial u}{\partial p_i} \right) dt$$

The idea is illustrated in Fig.6, where it can be seen that the total surface under the J curve can change continuously in spite of the discontinuities of s .



Similar analysis can be performed for other types of parameterizations. Concluding: for continuous input parameterizations, both in the value of the real decision

variables or in the time instants when the changes take place, it is safe the use of gradient based methods from the point of view of its computation.

Nevertheless, notice that if a binary input is considered, the gradient $\partial J/\partial p_i$ for this particular input, will not be defined, which stops us from a direct application of the methods. Only MINLP algorithms based on relaxation of the integer variables, such as Branch and Bound, could be applied. In the following sections we will see ways to overcome this limitation.

Many hybrid control processes belong to this class of problems. For instance, those with continuous dynamics but with a mix of continuous and on/off actuators, those with simultaneous batch and continuous processes which decision variables includes the times of starting or stopping a batch unit, etc. In all of them the hybrid character is linked to the mix integer / real nature of the decision variables more that to structural changes in the process.

2.2. State activated discontinuities

A second family of hybrid processes are those where the discontinuities are associated with structural changes of the process model that take place when some

variables reach some values. Many practical processes present these characteristics. For instance, phase changes associated to temperature and pressure, fluid regimen changes associated to velocity, etc. In this family, a value of the state variables activates the transition function, so that when it changes sign, the transition from stage k to $k+1$ takes place. See Fig.7:

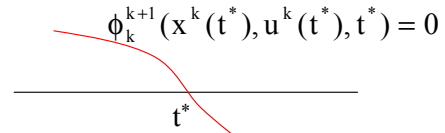


Figure 7

Notice that, in this case, even if the transition occurs without a jump in the states, that is, with a transition condition like (11) that assures the continuity of the states, the initial conditions of the new sensibility equations (12) after a transition are given by (13), which means that the time evolution of the sensibilities will present a discontinuity as shown in Fig.5.

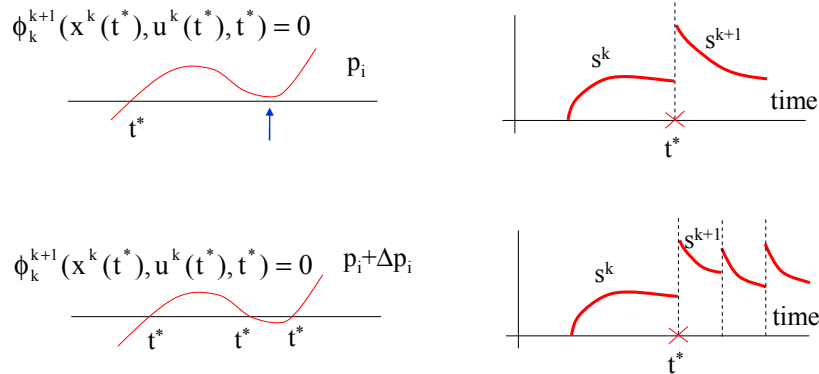


Figure 8

In principle, the situation is similar to the one described in the preceding paragraph, when the time of occurrence of the discontinuities was a decision variable, and, in principle, it should not stop us from using gradient based optimization methods. However, there is an important difference: In the input discontinuity case, the number and order of the discontinuities was fix, being implicit in the definition of the parameterization. By the contrary, in the state activated discontinuities, it may happens that a small change in a decision variable p_i modifies the number of times the transition function ϕ crosses the zero level, activating a different number of discontinuities and, hence, creating a discontinuity in J for this value of p_i so that the gradient is not defined. The idea is depicted in Fig.8 were a small change in p_i creates new discontinuities, which cannot be predicted in advance.

Notice that this means that, unless the number of discontinuities created by the state evolution in response to the decision variables is fix, the

sequential approach to optimization of hybrid system with gradient based methods, can face serious problems if the above mentioned phenomena takes place.

3 REFORMULATING SOME TYPES OF OPTIMIZATION PROBLEMS WITH HYBRID MODELS

Having in mind the real-time application of the optimization in the context of hybrid MPC, it is possible for some classes of hybrid systems to reformulate the associated optimization problems, so that, instead of a MINLP problem, a more efficient NLP one can be executed every sampling time.

We will discuss the reformulation focusing on the two types of hybrid processes that were mentioned in section 2, namely, those which hybrid character comes from the nature of the input and those with discontinuities activated by the time evolution of the states.

2.3 Hybrid processes characterized by input discontinuities

In the first group, we can mention two broad class of hybrid systems to which the reformulation can be applied:

1. Systems with on/off actuators, which value has to be computed along time in order to optimize a cost function.
2. Scheduling of batch processes operation, where start or discharge signals have to be activated from time to time in order to obtain an optimal sequencing.

Three key ideas are behind the reformulation:

- First, to formulate the problem in continuous time.
- Second, to switch from computing the (binary) values associated to a discrete variable in a set of predefined time intervals, to computing the time instants when changes in its values will take place, which are real values.
- Third, to embed the associated model changes in the simulation in the context of sequential approach, as in Fig. 1, applying a NLP optimization method.

Fig.9 gives a graphical description of them. The upper graph represents the usual way of describing the parameterization of a binary variable δ along time: a set of binary values δ_i can be assigned to different time intervals, typically the size of the sampling time. The optimization problem involves then the binary variables δ_i , one for every time interval, and has to use some MINLP algorithm. Instead, the time evolution of δ can be described as a series of pulses where the time instants at which the signal changes value, t_1, t_2, t_3, \dots in Fig.9, are real variables that can be used as parameters to be determined by the optimizer. This parameterization avoids the use of integer variables as decision ones, allowing the use of NLP algorithms in the optimization.

The previous discussion about the computation of gradients gives us some directions on when it is safe to use this approach. The main point to remember is that the number of discontinuities induced by the input parameterization must be constant, which translates into a constant number of “pulses”, like the ones of Fig.9, along the decision horizon. The assumption, then, is that the behaviour of the process requires the frequent change of its binary inputs, so that several pulses will be present in a reasonable period of future time. This type of systems is typical of batch process operation or in the case of on/off valves. This sub-class of hybrid systems can be denoted generically as “periodically actuated systems”. See (de Prada et al. 2008, 2009).

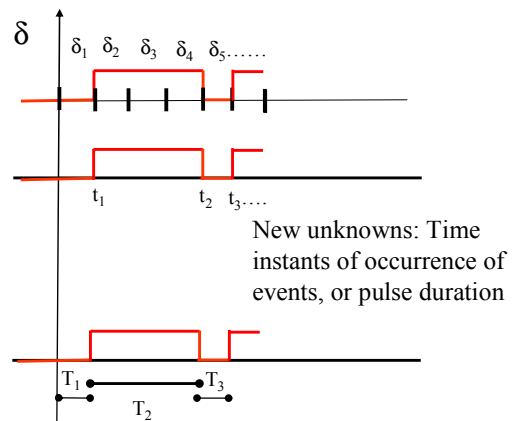


Figure 9

Notice that, a concept similar to “control horizon” in continuous MPC can be defined as in Fig.10, where the unknown duration of a certain number of the pulses (the “control horizon”) can be estimated by the optimization, forcing the following ones to be equal to the last one of the control horizon. This will save computation time and impose a certain regularity condition assuming that the process will operate under “steady” conditions with a given pattern.

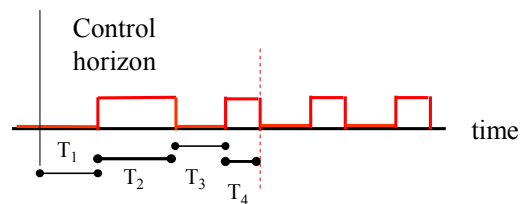


Figure 10

Nevertheless, it is not difficult to see that in case of input discontinuities where the number of pulses, that is, of state changes of the input variables, is not constant along the prediction horizon, the gradient can be evaluated incorrectly. This may happens with on/off variables that can change or not its state according to the operating conditions. Notice that this corresponds to values zero of some of the intervals T_i in the parameterization of Fig.10. as in Fig.11, where two pulses have been converted into one letting the optimizer to minimize down to zero an interval.

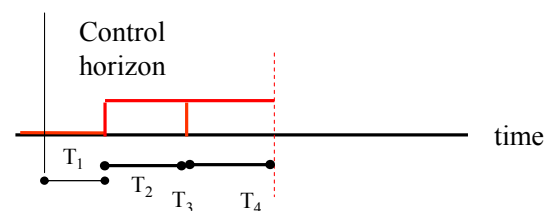


Figure 11

2.3 Hybrid processes characterized by state activated discontinuities

In the second group above mentioned, the hybrid processes with discontinuities activated by the time evolution of the states, the approach is similar:

Formulate the problem in continuous time and embed in the simulation the structural model changes that will appear as a result of the time evolution of the model for a set of values of the manipulated variables. The problem is solved in the context of sequential approach, as in Fig.2, applying a NLP optimization method. Notice that practical cases of processes with characteristics from both groups are frequent in practice. Conditions under which this approach is feasible will be examined in the following paragraph.

Again, the discussion above on the gradients gives us some clues on how to formulate the optimization of hybrid dynamical systems in order to avoid numerical problems, but this time is clear that only in a subset of this broad family can the sequential approach with embedded discontinuities be applied safely: The one of the processes where the number of discontinuities is kept constant.

Nevertheless, it comprises many practical cases³, in particular when starting up, or stopping down, a process unit, operation that implies a predefined set of procedures leading the process to the final state.

According to the previous discussion, any parameterization of the (possible) continuous manipulated variables avoiding numerical problems with the computation of gradients requires to respect the discontinuities created by the state variables, hence, it must be a “flexible” one made between these discontinuities as in Fig.12, where t_1 , t_2 , etc. are the time instants of the state event. So, in relation to Fig.12, the optimization algorithm will modify the values of u_1 , u_2 , etc. but the interval of application will be decided internally to the simulation by the state events.

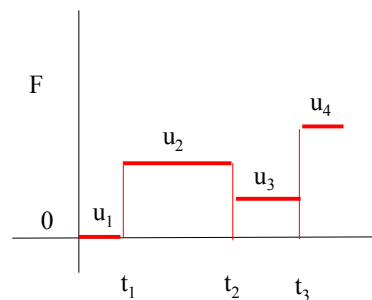


Figure 12

For other types of problems either a MINLP algorithm will be required making the problem more computationally demanded, or, in cases where only a small number of discrete decision is involved, a exploration of the different combinations can be explored.

4 SOFTWARE ENVIRONMENTS

The efficient implementation of the previous methods in the context of MPC requires a set of tools that are available in some modern modelling languages:

First, a simulation environment supporting hybrid models with a rigorous treatment of the discontinuities. The simulation of the hybrid process model will embed continuous dynamics and events and model changes activated either by state or input transitions.

Second, linking the simulation with an appropriate optimization software incorporating NLP methods. The joint simulation – optimization will provide optimal values of the decision variables for a given set of initial and boundary condition and should be encapsulated in a software component, such as the C++ class of Fig.13.

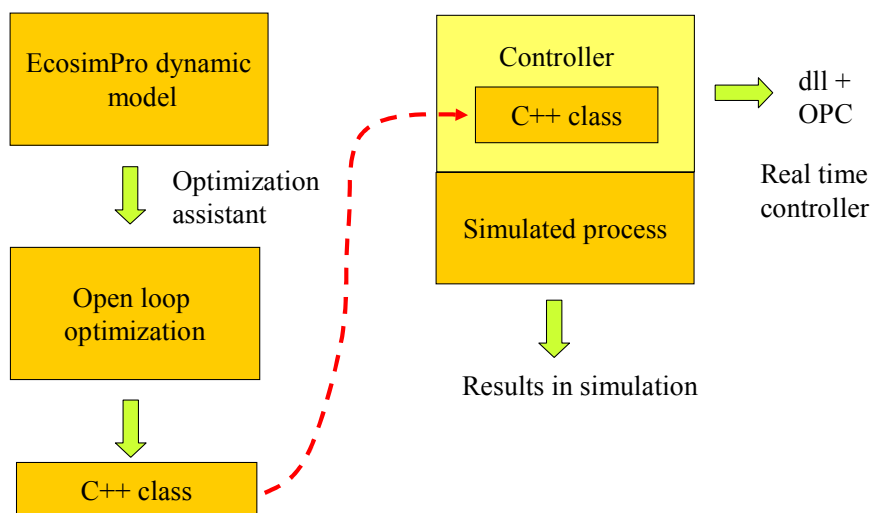


Figure 13

Last step is to embed the C++ class in a controller that calls it every sampling time to perform the control action according to the MPC philosophy.

In addition, external communications, such as OPC, will likely be required to connect the hybrid controller to the lower level control

system, operating in cascade over it on the process.

REFERENCES

- Bemporad, A. and Morari, M., 2000. Predictive Control of Constrained Hybrid Systems, *Nonlinear Model Predictive Control*, Ascona, Switzerland, 26, 71-98, (Progress in Systems & Control Theory) (Proceedings of International Symposium on Nonlinear Model Predictive Control - Assessment and Future Directions, June 1998)
- Floudas, C.A., 1995. *Non-linear and Mix-Integer Optimization*, Oxford Univ. Press.
- Galán, S., Feehery, W.F., and Barton, P.I., 1999. Parametric sensitivity functions for hybrid discrete/continuous systems. *Applied Numerical Mathematics*, 31, 17-47.
- de Prada, C., Grossmann, I., Sarabia, D., and Cristea, S., 2009. Non-linear Model Predictive Control of a mixed continuous-batch process. *Journal of Process Control*, 19, 123–137.
- de Prada, C., Sarabia, D., Cristea S., and Mazaeda, R., 2008. Plant-Wide Control of a Hybrid Process, *International Journal of Adaptive Control and Signal Processing*, 22(2), 124-141.
- de Prada, C., Sonntag, S., Engell, S., and Sarabia, D., 2009. Model-predictive control of hybrid systems: application to industrial case studies, *HYCON Handbook*, Cambridge University Press, in press.

AGENT SIMULATION OF FLOUR SILO INSTALLATIONS AS PLANNING TOOL FOR DECISION-MAKING

Alexander Ulbrich^(a), Ralf Kraul^(b), Christoph Tilke^(c), Mukul Agarwal^(d), Willibald A. Günthner^(e)

^(a) ^(b) ^(c) ^(e) Institute for Materials Handling, Material Flow and Logistics (fml), Technische Universität München, Germany
^(d) Corporate Technology, Bühler Management AG, Uzwil, Switzerland

^(a) ulbrich@fml.mw.tum.de, ^(b) kraul@fml.mw.tum.de, ^(c) tilke@fml.mw.tum.de, ^(d) mukul.agarwal@buhlergroup.com,
^(e) guenthner@fml.mw.tum.de

ABSTRACT

In this paper we discuss the approach of using simulation for the planning process of flour silo installations. After a brief introduction in the topic the usual performed planning process is presented. Advantages of using simulation as support tool, the requirements for the acceptance and the establishment of simulation during the planning process are described; the modeling of such a planning tool especially as agent simulation tool is discussed and a usual workflow of using the planning tool is illustrated. After that some possible results are shown and the applied validation techniques are briefly discussed. Last but not least an outlook is given.

Keywords: flour silo installation, planning tool, agent simulation, decision support system

1. INTRODUCTION

Flour silo installations assure the supply of a great number of various flours. On the one hand in these systems the basic flours are received from the flour mills and these flours are stored in large silos and are aged over several days. On the other hand there are produced a lot of further products through mixing of different basic flours. For instance depending on the production program there can be produced up to about 200 mixing products from about 20-30 basic flours. There are two different types of realizing the mixing process. One opportunity is mixing by mechanical conveyors. This was state of the art in former times because long-winded transport processes by pneumatic conveyors were too energy-intensive rather too expensive (Eberle 2009). Certainly for the mixing process by mechanical conveyors additional rearrangements of flour become necessary because all the flours to be mixed need to have access to the same mixing conveyor. Furthermore in this case of process the blending of small flour portion is only possible by mixing the product in multiple steps. A simplification of the production processes can be achieved by using batch mixers for the mixing process (Strauch 2003). This offers a more flexible, short time and faster order processing, so that normally a smaller number of silos

are needed in case of producing mixing products only in the moment of order execution without any intermediate storage. Moreover the sanitation requirements of flour silo installations become more and more important so that nowadays in new planning, restructuring and enlargements pneumatic conveyors are widely assembled instead of mechanical conveyors (Eberle 2009).

2. STATE OF THE ART

The planning of flour silo installation and mainly the identification of the expected capacity is a very complex challenge due to the scores of requirements and mutual blocking processes through the use of shared connection paths between silos, mixers and loading stations. Because static analytical methods of calculation provide only quite rough key figures, since dynamical correlations cannot be considered, the planner of such an installation normally relies mainly on his experience and offers solutions to problems which are known from the on-going production. To increase transparency of the planning process and therewith to assure the solution and increase the customer acceptance the Institute of Material Handlings, Material Flow and Logistics (fml) of the Technische Universität München in collaboration with the Bühler Management AG company developed a planning tool. This planning tool offers the agent simulation of flour silo installations with its complex dynamic processes.

3. ADVANTAGES OF USING SIMULATION

Using simulation for planning of flour silo installations brings along the following important advantages:

- planning is based on a substantially broader database
- dynamic dependencies in particular interactions between the individual components of the flour silo installation can be considered
- detailed investigation of several versions
- better understanding of the production processes

Simulation offers the possibility to consider changes in the system load and to compare several layout versions against each other. In flour silo installations a big number of jobs cannot be executed at the same time because they need one or more of the same resources as other jobs or processes. Furthermore specific jobs depend on other jobs if premixing-processes are required. Also aging times of the flours have to be regarded. In our opinion it is not possible to consider these dynamic dependencies by other means than simulation. Another great advantage is the larger detail of data output by simulation which allows the detailed investigation of several versions. Moreover the simulation analysis leads to a better understanding of the production processes.

4. REQUIREMENTS ON SIMULATION

First of all the main reason for the lack of use of simulation during the planning of flour silo installation is the high complexity of these systems and the high effort needed to built simulation. Due to this efficient modeling of such systems is needed for simulation. This can be achieved by using configurable standard components and general implementation. The implementation of the standard elements has to be so abstract that a broad market share is covered and finally almost all flour silo installations easily can be simulated by configuration.

In the same way the modeled processes of a milling plant have to be applicable for various layouts.

Because the succession of jobs is depending on the state of the silo installation no fixed succession is assumed. Therefore the approach is seen in agent simulation. An agent decides based on the state of the silo installation which process has to be performed next and how.

To increase the acceptance of the tool the planner has to be able to use the planning tool without or with less additional knowledge about simulation. Therefore all parameters and components have to be configurable through centralized tables.

Because this investigation needs no information about the flow characteristic of flour, the use of a material flow simulation environment as e.g. PlantSimulation (eM-Plant) is suitable. Therefore the bulk good has to be discretized to small piece goods.

5. MODELING OF THE SYSTEM

In the next paragraph we want to illustrate which elements have to be modeled for a planning tool used to design flour silo installations. First standard components and their attributes have to be identified and implemented so that the layout can be built up (section 5.1). For the control of the system agents with different goals have to be modeled (section 5.2). Additionally an automatic flow-path finder is needed to find possible conveyor combinations from source to sink to ensure operability with any layout (section 5.3) and a versions manager for different configurations is desirable (section 5.4).

5.1. Standard Components

The components of a flour silo installation are straightforward after increasing the level of abstraction. Besides mills for the flour supply, there are silos, conveyors, loading stations and batch mixers. Important attributes of mills are the number of different flours (vary in quality) and the capacity for each flour. Silos attributes for instance are the output capacity and the volume. Additionally every silo has to save the information of different layers in case of differently aged flour. The conveyors main attributes are the capacity and the variable mixable or not. Mixable in this context means mainly if the conveyor is mechanical and if mixing on it is allowed or not. Loading stations differ mainly in bulk and packing loading stations, have pre-bins and the capacity. Last but not least the batch mixers have a number of pre-bins and attributes for the batch size and the capacity.

5.2. Agents

For the control agents with different goals are necessary. In our planning tool an agent is defined as illustrated in figure 1.

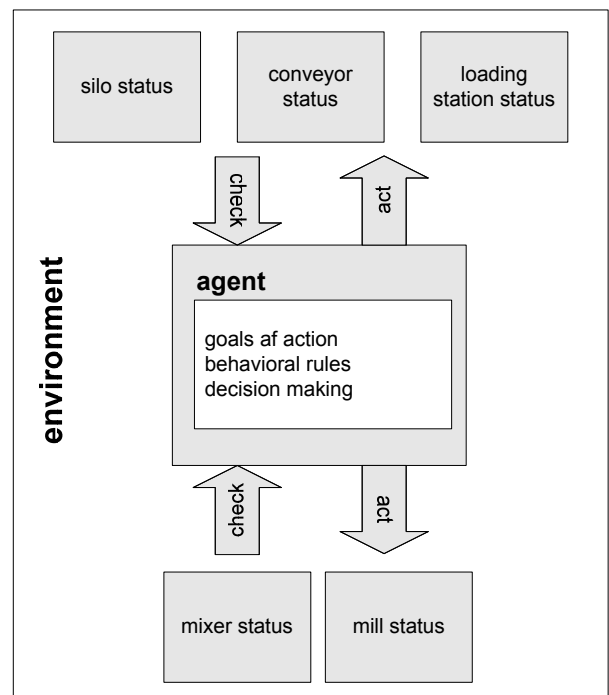


Figure 1: conceptual model of an agent

Every agent has goals depending on their tasks and observes the environment state (Macal 2008). The environment state in flour silo installation is mainly given through the status of silos, conveyors, loading stations, mixers, mills, material, jobs and orders. With knowledge of the abovementioned status an agent determines possible actions and evaluates them according to behavioral rules. In our planning tool there are four different agents identified and distinguished.

The first agent has to arrange that the mills have enough space/silos to produce at any time. Therefore it

must recycle the flour of specific silos to other silos which are not connected with the corresponding mill. These silos are generally assigned for recycling. This action must take place in silo installations where the space in silos connected to the mills is not sufficient.

Moreover an agent has to arrange that products requiring mixing by mechanical conveyors are transferred to silos which are connected to the same conveyor. In installations with more than one mill and with mixing products consisting of flours from two or more mills, recycle jobs from specific silos to others are unavoidable. Due to this not each silo can be assigned fixed with a particular flour. The agent has to permit feasibility of all requested customer orders.

For the task of order dispatching the agents three and four are identified, because order dispatching can be distinguished in two sections. Agent number three's goal is to start requested orders. These can be the following tasks:

- flour from one silo to a loading station
- flour from several silos transported via mixing conveyor to a loading station
- flour from several silos transported via mixing conveyor to a silo
- starting orders mixed by batch mixers and initiating filling jobs for the batch mixers
- flour from one silo to a pre-bin of a batch mixer according to the requested filling jobs

The goal of the last agent is the order processing in the batch mixers. It has to check which jobs can be performed and which sequence should be chosen. The feasibility is depending on the availability of silos or loading stations, the availability of enough flour in the pre-bins and the availability of the batch mixer.

5.3. Automatic path-way finder

Due to the fact that conveyors sometimes have more than one in- or/and output stream there are a big number of possible path-ways from any source to any sink. Since the planner or user of the planning tool only defines the connections between all components via tabulations, an algorithm had to be implemented to identify every possible path-way. For this a list is generated and loaded with the gathered information. Therewith the agents can quickly identify possible path-ways.

5.4. Versions manager

During the work with the planning tool bottlenecks can be identified in the designed versions. Due to this often from initially two or three versions a big number of different versions grow up. But changes from version to version differ only in a small number of configuration variables. Hence it is desirable that the configuration is saved separately and can be reloaded into the actual release of our planning tool. Due to this the configuration tabulations are centralized and saving or loading is done by a versions manager.

6. WORKFLOW OF PLANNING

The workflow of planning with our tool can be divided in to layout configuration of the flour silo installation, specification of the material flow system, definition of data needed for the information system, the simulation test runs and finally the interpretation of results.

6.1. Layout configuration

According to figure 2 the planner has to carry out the following steps to configure the layout of the flour silo installation.

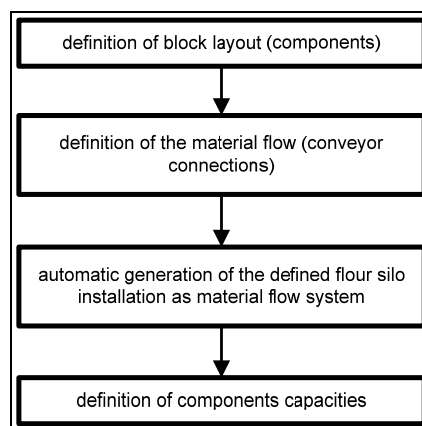


Figure 2: workflow of layout configuration

First of all the planner has to define the number of components needed to describe a layout version and the approximate placement inside a specified grid. This can be done by filling a table with key words. The result of this step is an unassociated scheme of the installation containing the defined number of mills, silos, loading stations, mixers and conveyors.

In the next step the connections between the several components have to be configured by definition of input and output streams of every conveyor. This step defines the whole material flow and a downstream algorithm can detect every possible pathway. Afterwards the whole material flow system is drawn and components capacities can additionally be defined in centralized configuration lists.

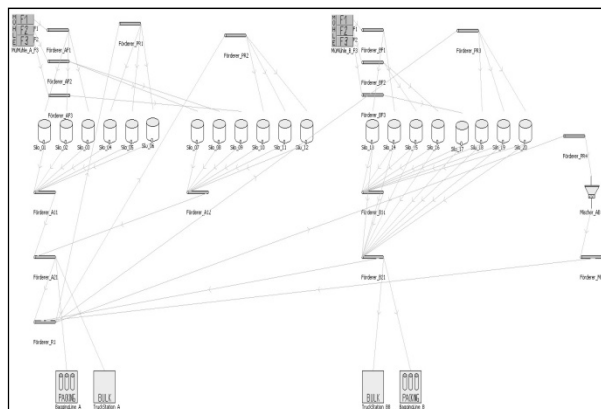


Figure 3: example of a configured layout

In figure 3 an example for the configuration of a flour silo installation is illustrated.

6.2. Data input

The workflow for defining the information system of a flour silo installation is illustrated in figure 4.

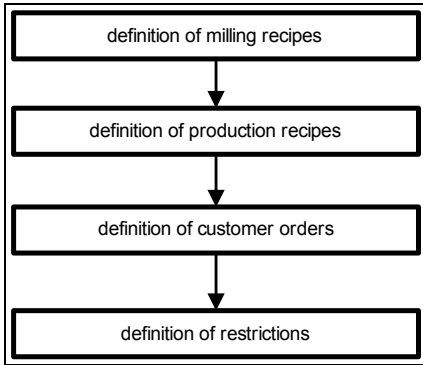


Figure 4: configuration of the information system

There are several data lists needed to describe the information system of a flour silo installation. Besides the customer orders the recipes for mixing production, the milling recipes for the basic flours, the dependencies between basic flours, the assignments of basic flours to mills and some additional restrictions are needed. Usually the planner performs the data input according to figure 4. Restrictions can be that some mixture recipes can only be performed on special components (e.g. mixers).

6.3. Test runs

After configuration is made by the abovementioned steps the simulation test runs are performed automatically and several time stamps are collected. By running the simulation slowly the planner can inspect the correctness of processes by animation of the material flow streams. Simulation of a big flour silo installation with more than 100 Silos, several loading stations and 2-4 mixers takes about an hour runtime for a customer order list containing all work of a month.

6.4. Outputs

Finally there are a big number of possible key figures. Some of them are shown below. In figure 5 for instance the layout of the flour silo installation is illustrated in style of a sankey diagram. Often used connections are marked according to the quantity of material flow.

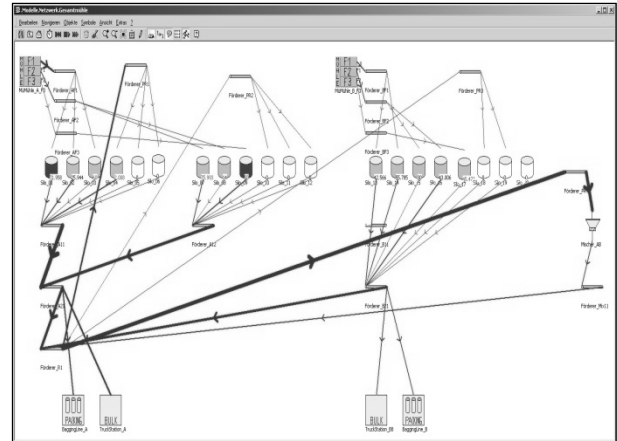


Figure 5: visualization after a test run

Because the number of orders differs from day to day, the workload per day is illustrated in another diagram. Figure 6 illustrates the quantity of completed orders per day of an example installation.

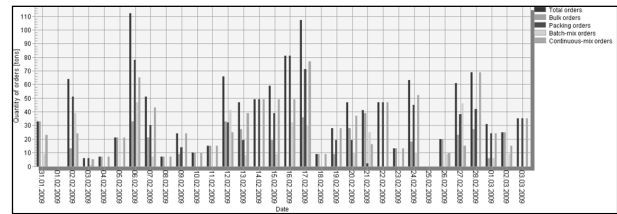


Figure 6: quantity of completed orders per day

Often the aging-time is another quality benchmark of the planning results. Normally there is a minimum time (hard requirement) for which the flour must be aged and a desired time (soft requirement) for which it should ideally be aged. Figure 7 shows the aging time of the completed orders of an example.

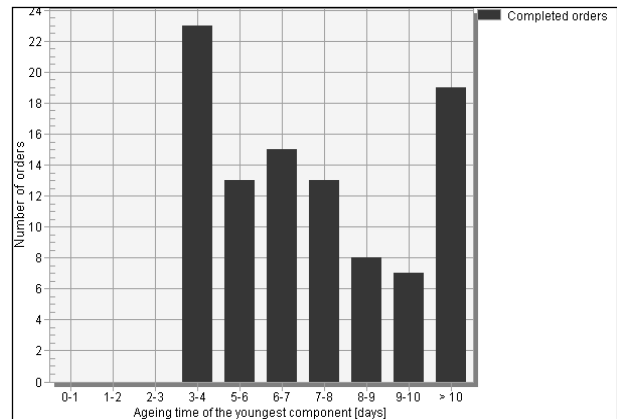


Figure 7: ageing time of completed orders

Furthermore a focus on recycling activities in flour silo installations is applied. This is on the one hand to save energy costs and on the other hand to free silos and conveyors by saving recycling jobs. The number and quantity of recycling jobs depends mainly on the layout and the preferred mixing technique. To visualize these

processes another diagram in Figure 8 shows the quantities of flour transfers from and to silos.

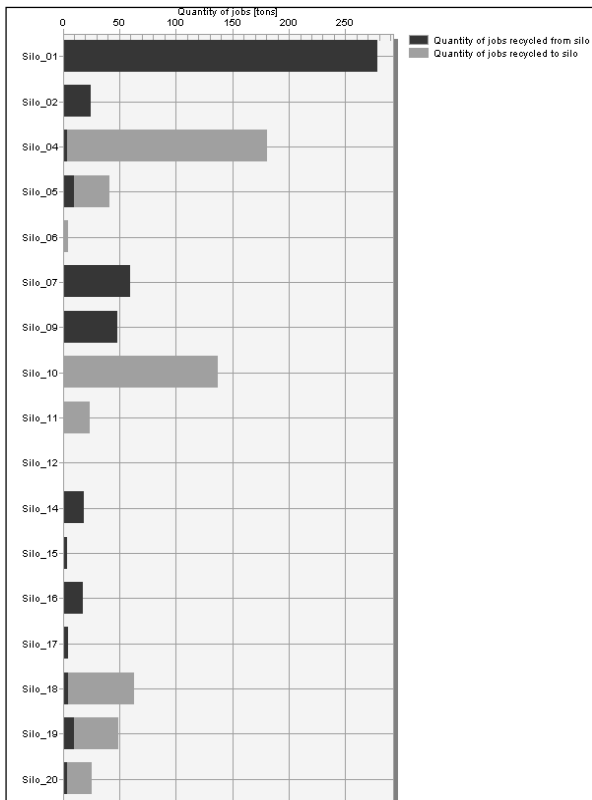


Figure 8: quantity of recycled jobs from/to each silo

7. VERIFICATION AND VALIDATION

There are several approaches discussed in literature to decide whether a simulation model is valid or not (Sargent 2007, Schlesinger 1979). Three of these are applied for this model. The first one is subjective and performed by ourselves. Therefore we observed some different customer orders if they are performed as desired and evaluated the results. In the next step, the validity of the simulation model had to be decided by the model users. Another approach used to decide the validity was to compare the results of our planning tool with the results of an existing silo installation in the same time period. Therefore data of a customer of Bühler AG were used and a big model was configured. The results achieved by the simulation could then be compared to recorded data from the real installation.

All performed approaches were successful and affirmed the validity of the simulation model.

8. CONCLUSION AND OUTLOOK

By the agent simulation of flour silo installations the planning of such systems becomes more transparent and the acceptance by the client is increased. Till now there is no better way to evaluate different planning versions. The requirements of these systems can be considered and the versions can be compared to each other.

In future it is no more imaginable for us to plan a flour silo installation without using such a planning

tool. Moreover the simulation tool can be used in other branches as e.g. the animal-food industry (Kersten, Rohde and Nef 2005).

REFERENCES

- Eberle, S., 2009. Mehl mischen oder homogenisieren - Umbau von vorhandenen Silo- bzw. Mischereianlagen. In *Mühle + Mischfutter*, 146. Jahrgang. Volume 3. February 2009, pp. 73-81, ISSN 0027-2949. Moritz Schäfer Verlag, Detmold, Germany. (in German)
- Macal, C., North, M. 2008. Agent-based modeling and simulation: ABMS examples. *Proceedings of the 2008 Winter Simulation Conference*, pp. 101-112. December 7-10, Miami, Florida, USA.
- Kersten, J., Rohde, H.R., Nef, E., 2005. Mischfutterherstellung: Rohware, Prozesse, Technologie. AgriMedia, Clenze, Germany. (in German)
- Sargent, R.C., 2007. Verification and validation of simulation models. In *Proceedings of the 2007 Winter Simulation Conference*, pp. 124-137. December 9-12, Washington D.C., USA.
- Schlesinger, et al. 1979. Terminology for model credibility. *Simulation* 32 (3). pp. 103-104.
- Strauch, W., 2003. Marktübersicht Feststoffmischer. In *Mühle + Mischfutter*, 140. Jahrgang. Volume 24. November 2003, pp. 721-725, Moritz Schäfer Verlag, Detmold, Germany. (in German)

AUTHORS BIOGRAPHY



Dipl.-Inf. Alexander Ulbrich was born in Munich, Germany and attended the Technical University of Dortmund, where he studied Applied Computer Science with Mechanical Engineering and obtained his degree in 2005. Now he works as a researcher at the Institute for Materials Handling, Material Flow and Logistics, Technische Universität München (www.fml.mw.tu-muenchen.de) to obtain his conferral of a doctorate in engineering.



Dipl.-Ing. Ralf Kraul was born in Munich, Germany and attended the Technische Universität München, where he studied Mechanical Engineering and obtained his degree in 2004. Now he works as a researcher at the Institute for Materials Handling, Material Flow and Logistics, Technische Universität München (www.fml.mw.tu-muenchen.de) to obtain his conferral of a doctorate in engineering.



Dipl.-Ing. Christoph Tilke was born in Wolfratshausen, Germany and attended the Technische Universität München, where he studied Mechanical Engineering and obtained his degree in 2004. Now he works as a researcher at the Institute for Materials Handling, Material Flow and Logistics, Technische

Universität München (www.fml.mw.tu-muenchen.de) to obtain his conferral of a doctorate in engineering.



Dr. Mukul Agarwal is responsible for corporate-wide research and development in Intelligent Process Operation at Bühler Management AG (www.buhlergroup.com). He has co-authored over 70 papers, about half of them in refereed journals. Prior to joining Bühler AG in 1999, he was a senior lecturer at the Swiss Federal Institute of Technology, Zurich, since 1989.



Prof. Dr.-Ing. W.A. Günthner is the head of the Institute for Material Handlings, Material Flow and Logistics, Technische Universität München (www.fml.mw.tu-muenchen.de) since 1994.

AN AGENT-DIRECTED MULTISIMULATION FRAMEWORK FOR MANAGEMENT SIMULATION GAMES

Galina Merkurjeva^(a), Jana Bikovska^(b), Tuncer Ören^(c)

^{(a),(b)} Department of Modelling and Simulation
Riga Technical University
Kalku Street 1, LV-1658 Riga, Latvia

^(c) The McLeod Modeling & Simulation Network (M&SNet) of SCS
School of Information Technology & Eng. (SITE)
University of Ottawa, Ottawa, Ontario, Canada

^(a)Galina.Merkurjeva@rtu.lv, ^(b)Jana.Bikovska@rtu.lv, ^(c)Oren@site.uottawa.ca

ABSTRACT

The paper presents an overview of agent-based multisimulation applications in management simulation games. The scenario-based game management concept is introduced in the paper. Agent-directed multi-simulation framework for management simulation games is developed. An illustrative example for a specific web-based logistic simulation game is provided.

Keywords: multisimulation, multi-models, agent-based simulation, agent-supported simulation

1. INTRODUCTION

Computer simulation games are widely used in management education to provide extensive experience under control conditions. In general, they are contrived situation which imbed players in a simulated business environment, where they make management-type decisions from time to time, and their choices at one time generally affect the environmental conditions under which the subsequent decisions must be made. In other words the main purpose of management simulation game is to simulate interactions between business actors, so they are well suited for using agent-directed approaches. Multisimulation is a novel simulation methodology allowing simultaneous explorations of different simulation scenarios which plays very important role within the structure of simulation games.

Agent-directed simulation that integrates agent and simulation paradigms has two distinct categories (Ören, 2000; Yilmaz, 2005; Yilmaz and Ören, 2006): (a) use of simulation for agents or agent simulation, and (b) use of agents for simulation (agent-supported simulation and agent-based simulation). Agent simulation, i.e., simulation of agent systems is used in several application areas: manufacturing systems, robotics, transportation, logistics, e-commerce, etc. Agent-supported simulation involves the use of intelligent agents to improve

simulation and gaming infrastructures and environment. Agent-based simulation is the use of agent technology to generate or monitor model behavior in a simulation study. In this study, in addition of agent simulation, agent-supported and agent-based simulation is used for the development of management games.

2. AGENTS IN TUTORING SYSTEMS

Agent technology is widely used in different tutoring systems. For, example, in (Hospers et al., 2003) such an agent-based system for nurse education is discussed. Here authors describe the development of an effective teaching environment called Ines or Intelligent Nursing Education Software that uses agents to support learning and is able to provide sensitive feedback, demonstrations of practical exercises and/or explanations. Two architecture styles are used in Ines: (1) "data abstraction and object-oriented organization," and (2) "event based, implicit invocation." The first one means that single components are represented by objects that have their own attributes and methods. Objects could be identified as physical objects used by nurse in practice (needles, swaps, etc) and agents used in learning process. Objects are implemented by using several classes within Java. The second architecture style is used to implement agents' behaviour in the tutoring system. Each agent is capable of transmitting messages to other agents, and receiving messages by observing all messages that are sent to that agent or that are broadcasted to all agents.

Ines architecture consists of four main parts: input from devices, a concrete user interface, an abstract user interface and the cognitive part which includes student, instructions, domain, and patient models. Two types of software agents, i.e. proactive and reactive agents, are used in the instruction model to provide teaching instructions to a nursing student. Each proactive agent observes the user interface and send an observation to a reactive agent. Reactive agents receive the observations

and respond to them. The following are examples of proactive (a) and reactive (b) agents that define their function, i.e., (a) Time Agent, Error Agent, Task Observer and Sterile Agent, and (b) Feedback Agent, Interaction Observer Agent and Examination Agent. To implement agents, the agent platform developed by the Parlevink group is used.

Capuano et al. (2000) present an innovative architecture for Agent Based Intelligent Tutoring System (ABITS) by using the Multi-Agent paradigm. ABITS extends a traditional Course Management System (CMS) with a set of intelligent functions which allow student modeling and automatic curriculum generation and aimed to improve the learning effectiveness based on the adaptation of the teaching material to student skills and preferences. ABITS knowledge indexing is based on metadata about the learning objects and their interrelations, and conceptual graph that is used to link concepts underlying the knowledge domain, i.e., prerequisite, sub-concept, and so on. To implement a multi-agent paradigm in ABITS, Java-based Agent framework JAFMAS is used. ABITS implements three main types of agents, one for each base function topology. Evaluation Agents function is evaluating and updating of the Cognitive state that contains the knowledge degree, reached by a particular student. Affective Agents perform evaluating and updating Learning Preferences that enclose information about the student perceptive capabilities. And Pedagogical Agents perform evaluating and updating Course Curriculum depending on learning goals, cognitive states and learning preferences. Additionally, to ensure communication between CMS and ABITS, Spooler Agent is embedded in the CMS module. When CMS requires an ABITS service, it requests this service to the Spooler Agent which requests services from Evaluation, Affective or Pedagogical agents as needed. The main innovative issues of ABITS system are based on adoption of software agents and distributed nature of the software architecture.

3. AGENTS IN MANAGEMENT GAMES

The literature review provides several examples of agents' functions in management games. For instance, in Van Luin et al. (2004), the classic supply chain simulation and management game called, as the Beer Game, is remodelled as a multi-agent based simulation using the Agent-Object-Relationship (AOR) modelling language. Authors extend and refine the classical discrete event simulation paradigm by enriching it with the basic concepts of the AOR metamodel. The resulting simulation method is called as Agent-Object-Relationship Simulation (AORS). In AOR modelling language, an entity is an agent, an event, an action, a claim, a commitment, or an ordinary object. Agents can communicate, perceive, act, make commitments and satisfy claims while objects are passive entities without such capabilities. The state of AORS system contains the simulated time, the environment state including the non-

agentive environment (as a collection of objects) and the external states of all agents (e.g., their physical state, their geographic position, etc.), the internal agent states (e.g., representing perceptions, beliefs, memory, goals, etc.), and a list of future events. At the Beer Game four agents with similar behavior can be identified: the retailer, wholesaler, distributor and factory. Each agent receives orders from the downstream supply chain node and deliveries from its upstream node. Here, the incoming and outgoing orders are modelled as messages, the shipping of beer as a non-communicative action event and the reception of beer is modelled as non-action event. The reactions of agents to the incoming events are specified in the form of production type rules. For instance, the rule that is triggered periodically at the end of week (event) compares the current inventory with outstanding orders and either send the entire order or ship the maximum available quantity of beer. AORS model of the Beer Game have been used for testing different replenishment policies in supply chains, however authors did not consider how a supply chain agent could improve its performance by learning from the game past experience.

Overviews of agent based paradigms and enterprise simulation for e-learning and knowledge transmission is given by Remondino (2007, 2008). Author discusses the integration of cognitive and reactive agents to web based business games. Cognitive agents' behaviour is goal-directed and reason-based, so they may have different roles in the game, for instance, the agent can compute some strategies to be used to make profit in the simulation. That said, this agent could then be used both as a decision support system for human users – since it could foresee some results, based on its acquired experience – and as a virtual tutor, explaining the relations among certain variables (decisions) and the achieved results. Reinforcement learning algorithms are used to evaluate utility of action to take next. Reactive agents simply retrieve preset behaviours. From a structural point of view, the reactive agents can constitute some parts of the model itself, in order to make it self adaptive when facing unforeseen decisions and context (simulate customer or supplier, represent production plants, etc.). Evolutionary algorithms (for example, Genetic Algorithm) may be applied for reactive agents to solve action selection problem.

Dobson et al. (2004) investigated technologies, which may have significant impact on the business game-based training and as the most promising the agent technology is considered. Several possible functions of an agent in the business strategy game are proposed: (1) representing market entity, sometimes completely replacing a human player or working in the background as an individual assistant to the learner; (2) monitoring trainee and evaluating his/her actions online; (3) modeling consumers' behavior. Authors successfully implemented agent technology in strategy business game for the telecommunication industry. The developed game provides the assistance for learners via agents and the

following advantages are found: agents demonstrate optimal behavior by making rational decisions that offers a range of new opportunities for competency development in the trainees; the ability of the rule-based systems to provide the logics leading to decisions made by the agents for the human player allow the trainee to learn from these explanations in the process of the game; if necessary, the agent-based game can be run at much faster pace that offers an opportunity to run multiple scenarios in just one class session and in such way to investigate the impact of various parameters on the business and industry as a whole. However, all suggestions made by authors have declarative nature so more information is needed on particular models and agent realization.

The development of a simulation game, REAL Business, using an agent-based approach is described by Bai et al. (2007). The proposed architecture extends the traditional ICAI (Intelligent Computer Aided Instruction) architecture, which utilize the techniques of expert systems and artificial intelligence to help a person learn interactively and consists of expert module, pedagogical module, and user model. REAL contains the following components: (1) reflective agent which stores specific information about individual learners (e.g., level of competence) to be used by the pedagogical agent, (2) expert agent which contains expert knowledge of a particular subject domain (e.g., in REAL Business, the domain knowledge is on probability and in REAL Plant - ecology), (3) pedagogical agent which compares the knowledge of an expert agent with the reflective agent and thus provides teaching strategies (for instance, for those who showed competence within the first business community, the agent may generate situations where some resources are limited), and (4) communication agent controls the human computer interaction, it observes the user's mouse click and provides users with help for using tools; clarifies learning goals; gives navigation orientations, etc. In REAL business, Students are supposed to help a store owner design business strategies to run an ice cream store. They taught a reflective agent through observing the event network, probabilities to multiple events, as well as data from the prior sales and design production rules. The outcome is observed in the simulation game by evaluating the agent's performance. Data are collected, reorganized, and displayed for students so they can justify and evaluate their predictions based on the simulation and data evaluation. To implement agents the rule-based approach is adopted and JESS (Java Expert System Shell) is chosen as an engine. REAL is embedded in a rule-based system. A rule-based system consists of facts (agent's goal, belief, desire, physical conditions, etc.), rules (procedural knowledge, instructing how an agent should act at certain conditions), and a reasoning engine that acts on them. In the conclusion, author indicates that, although, pedagogical, reflective, and expert agents are not new, their integration of all three agents within an

educational simulation is unique and built framework for developing intelligent agents in simulation games.

Multisimulation is a novel simulation methodology which allows simultaneous explorations of different simulation models and/or scenarios. It was conceived by Ören (2001). Ören and Yilmaz –who also advocate agent-directed simulation (Ören, 2001; Yilmaz, 2005, Yilmaz and Ören, 2009)–contributed to multisimulation (Yilmaz et al., 2006, 2007; Ören and Longo, 2008). This article is the first one to use multi-simulation and agents in simulation games.

4. AGENT-DIRECTED FRAMEWORK FOR MANAGEMENT GAME

4.1. Scenario-based Game Management

The game simulation could be performed according to a predefined scenario chosen (or specially elaborated) by the game manager. Scenario plays very important role within the game structure and it has to be formalized for further agent implementation.

In general, a scenario can be conceived as a means of transforming the system's initial state to a final state, following main development trends, which are influenced by both internal events and external activities. It could be presented by the following 5-tuple:

$$\langle SI, T, E, A, SF \rangle,$$

where:

SI presents *initial state* of the system;

T presents predefined system development *trends* (e.g., demand process);

E presents predefined internal *events* (e.g., changes in demand process);

A presents performed external *activities* (e.g., number of produced items);

SF defines *results* or the system final state once the scenario has terminated.

The generic scenario structure is shown below (see Figure 1).

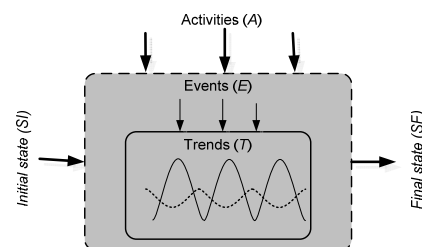


Figure 1: Generic Scenario Structure

Trends T_i introduced to model system dynamics over time t are defined by different functions, e.g.

$$T_i = f_i(t), T_i \in T.$$

Patterns of these trends depend on the type of function and its parameters.

All internal and external influences depend on the current system state and they are aimed at achieving particular result of the system functioning. Any internal and external influences lead to short-term or long-term changes in the pattern of trends, i.e.

$$f_i^E : E \rightarrow T_i \text{ and } f_i^A : A \rightarrow T_i,$$

where $E = \{e_1, e_2, \dots, e_n\}$ is a set of internal events; and $A = \{a_1, a_2, \dots, a_n\}$ is a set of external activities.

System initial state is defined by the system structure as well as by value of the trends' functions at the moment when the system investigation starts:

$$SI = T_i(t_0),$$

where t_0 defines the initial time instant.

In terms of management simulation game, scenario specifies the initial state of game's business environment and determines the development of the game situation through time according to predefined trends as well as contains the list of important events taking place during simulation. Events are indicated by the game manager and can affect trends. Participants make their decisions (or in other words, take actions) that can affect both trends and events. Trends may be configured according to the teaching goals by changing appropriate parameters available in the game.

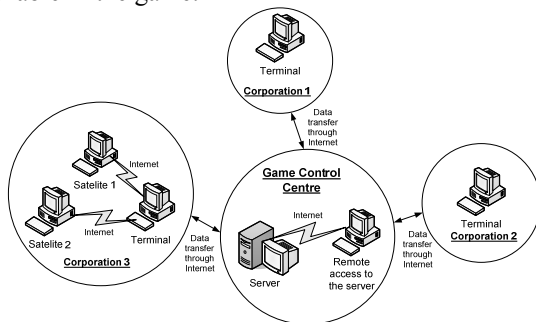


Figure 2: Game Software Structure

The structure of the game software is given in Figure 2 and it can be applied to the example below. The game software consists of two parts: (1) *Game Control Centre*, which performs function of the game server, where business environment is simulated and (2) *Corporations* that ensure user interface for entering various decisions. Decisions, entered by participants are transferred through Internet to the Game Control Centre then are processed there, and their consequences in the form of different reports are transferred back to terminals. Corporations can be located at a far distance from the Game Control Centre; since all communications are performed via Internet. The game manager can view the data of all participants and apply necessary scenario alterations. Additionally, corporations can run satellite terminals for

distributing decision authorities among corporation members as well as the game manager can have remote access to the game server from additional terminal.

The game is functioning in real-time mode. Decision-making and having results from the decisions made are provided in interactive real-time mode as they are in the real-world environment. Interactive mode means that decisions are made continuously when in the game model and game market situations occur which need to be reacting to by the participants. In the interactive model decisions are made as soon as they are needed or at least as soon as the decision-maker notices that the market situation needs actions from him.

4.2. Roles of Intelligent Agents

Intelligent agents can be introduced in the game in different ways (see Figure 3): (a) agents may perform tasks of the game manager related to the elaboration of the game scenario and control the game session, (b) play a participant role, (c) and perform a tutor function in the game.

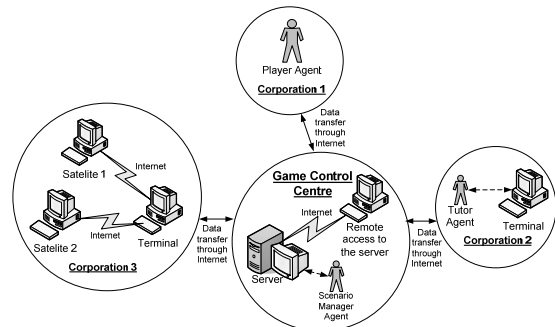


Figure 3: Agents in the Game Software Structure

4.3. Agent-supported Game Manager

An agent can perform tasks of the game manager (see Figure 4) related to the elaboration of the game scenario, i.e., to define and set initial conditions of the game according to a composed story or developed case study, monitoring the game session by checking trends and defining events, and making necessary changes in scenario during run-time by modifying trend functions, i.e.

```

set SI
for each j
  for each i
    define  $T_i(t_j)$ 
    generate  $e_i \in E$ 
    if  $e_i \neq \emptyset$ 
      modify  $T_i(t_{j+1})$ 
    end
  end
end
end

```

It is well-known from experience that the human factor significantly can create inconsistencies in the designed scenario which, with a high probability, may lead to serious mistakes in the game. Agents would have the capability to increase the consistency, completeness and other qualities of the game scenario and the reliability of its software. Production rules embedded in the game

software can be used to set up parameters of the business environment according to defined policies. In order to generate production rules, inductive learning algorithms can be used. Monitoring of the game scenario is based on the dynamic comparison of parameters of the business environment and state of the companies in the game.

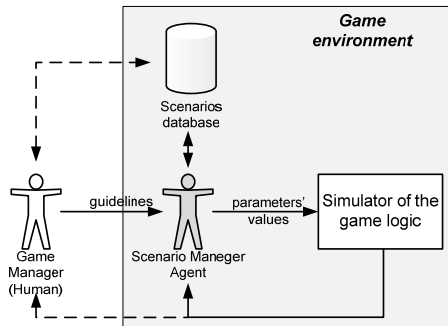


Figure 4: Scenario Manager Agent

4.4. Agent as a Virtual Player and/or a Tutor

If a business game of this type is used for individual training, the agent could substitute another competitive company and be introduced to simulate its behaviour, i.e. in the above algorithm, which describes behaviour of the Scenario Manager Agent, instead of event e_i , activity a_i is introduced. The structural scheme of an agent-based simulator is given in Figure 5.

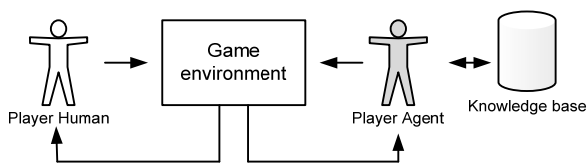


Figure 5: Virtual Player within the Game

When the agent plays a tutor role, it may define weak points in the players' activities, give necessary explanations in the form of causal relationships, and may suggest possible ways to solve the problems, if the participants need any support in their decision analysis. If the number of participants is large, it may not be so easy to manage the game due to its online mode and lack of time to make a deeper analysis of the current situation, i.e.

```

for each j
  for each i
    define  $T_i(t_j)$ 
    generate  $a_i \in E$  //player generates  $a_i$ 
    evaluate  $a_i$  //evaluate performance of
activity
  if performance  $a_i$  is not satisfactory
    generate  $a_i' \neq a_i$  //agent generates  $a_i'$ 
    modify  $T_i(t_{j+1})$ 
  else
    modify  $T_i(t_{j+1})$ 
  end
end
end
end

```

The corresponding scheme of Tutor Agent and Trainee interactions is presented in Figure 6.

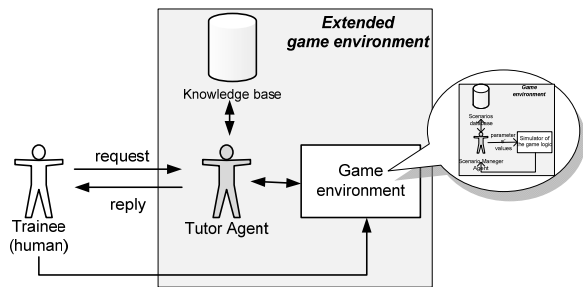


Figure 6: Tutor Agent and Trainee Interaction Scheme

5. MULTISIMULATION FOR GAME SCENARIOS

In general, multisimulation is a simulation, where at decision points, simulation updated may include decisions on branching of simulation studies as well as selection of models/submodels and associated parameters and experimental conditions to be analysed simultaneously and used in subsequent stage of simulation (Yilmaz et al., 2006). Therefore, from the above definition we can presume that multisimulation concept could be successfully applied for management of simulation games. Multisimulation methodology is introduced to enable exploratory simulation using various types of multimodels. A multimodel consists of several submodels where only one or some of them is/are active at a given time. The "multimodel" formalism introduced by Ören (1987, 1991) can be applied to specify scenarios.

5.1. Scenario Multimodel

Most of scenario management functions are described above. One of them is monitoring of the game session, and making necessary scenario alternations during run-time. Monitoring is aimed at managing key indicators of functioning of business environment. If any of indicators is out of critical value then the necessary changes should be introduced to the current scenario or, in other words, the alternative scenario should be applied that allow changing current situation according to the training goals. A multimodel is associated with a dynamic scenario that defines the sequence of scenarios in specific time moments (see, Figure 7), where S_i is the initial scenario in the game, but S_1, \dots, S_n – possible alternative scenarios, $S_1, \dots, S_1.m$ and $S_n.1, \dots, S_n.k$ – respective subsequent scenarios.

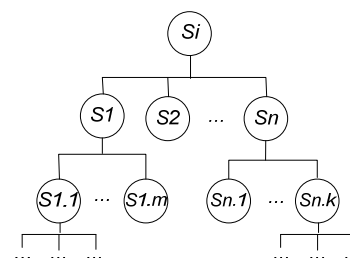


Figure 7: Scenario Multimodel

According to classification of multimodels introduced by Yilmaz and Ören (2004); Yilmaz et al. (2007), here single aspect multimodels (i.e., only one submodel is active at a given time) with static structure (i.e. model

does not allow structural changes) and adaptive behaviour are introduced. In the last case constraint-driven multimodels are controlled by stationary or adaptive transitions policies that can have learning capabilities. Therefore, the task of alternative scenario choosing can be delegated to the agent.

5.2. Agent-Supported Scenario Multisimulation

Choosing the alternative scenario requires computational support to (1) **observe** simulation state, (2) **reason** to qualify scenario for update, and (3) **plan** for constraint-driven activation by exploring potential paths within the state space of the problem domain. Each sub-scenario is viewed as an operator that transforms the state of the simulation model. Agent paradigm provides the necessary computational infrastructure to attain these objectives.

The scenario manager agent decouples the simulator of the game logic from scenario models and it is aimed at performing dynamic scenario updates. The activation conditions of submodels are defined in terms of production system. A hypothetical submodel activation process is shown in Figure 8.

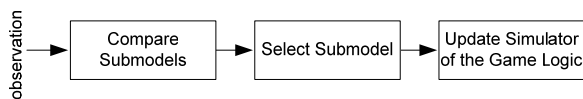


Figure 8: Submodel Activation Process

The input to the process refers to the observed condition used by the compare submodels component. Dynamic submodel replacement requires monitoring simulation conditions to determine if any of the potential state variables of interest are changed and it could be an indicator of a scenario update in the simulator. During the model comparison phase all submodels, the preconditions of which are implied by the current observed state are identified and their comparison is performed by the scenario manager agent. During the next phase only one submodel should be qualified for further exploration. Finally, simulator of the game logic is updated by the selected submodel.

6. AN ILLUSTRATIVE EXAMPLE

As an example of a management simulation game for introducing agent-directed multisimulation technology, the International Logistics Management Game (ILMG) is selected (Grubbström et al. 2005). The game is aimed at providing virtual business environment for training decision-making skills in the area of international logistics, production management, finances, etc. The game can be classified as a total enterprise (or general management), interacting, computer-aided and internet-based, real-time processed business simulation. ILMG software functioning according the structure presented in Figure 2.

Different scenarios in the game are introduced to simulate real world situations and force trainees to

acquire skills and experience in managing different functions of the company in various situations. During the game, participants operate as different corporations within international market, producing goods or services (i.e. making strategic and operational decisions) and competing each other in order to improve their corporation performance and to achieve the following goals: earn profit, capture substantial market share, and achieve high customer satisfaction.

The ILMG scenario is generated by the game manager according to the needs of a specific training goal and the participants' knowledge background. The business environment and the MIS reports are made as complex as in a real life or it could be simplified as necessary. Initial conditions of the game and corporations' state are set up by changing different parameters that can be divided to several groups (see, Figure 9): regional parameters (e.g., unit projecting cost, basic wage level), regional-related production parameters (e.g., productivity parameters, overheads parameters), technical coefficients of products (e.g., material per unit, setup cost), market related parameters (e.g., price effect parameters), financial parameters (e.g., prime rate, tax rate), etc.

Most of the parameters have to be defined before the game starts and they can't be changed afterwards. However some parameters (i.e., prime rate, business cycle index of the region, regional productivity index, and regional wage level) can be easily edited during the game that allows changing business environment to simulate different situations.

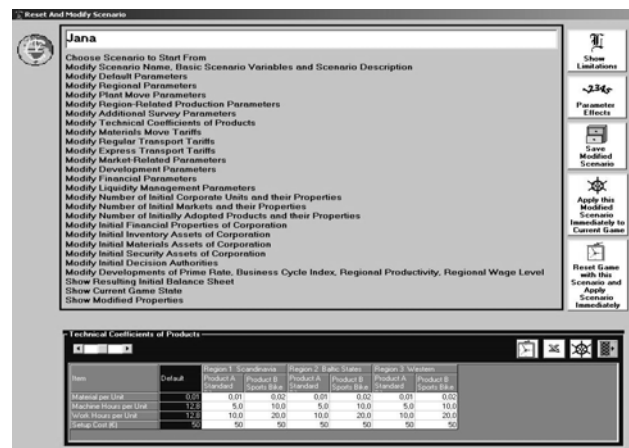


Figure 9: Scenario Elaboration Screen

In ILMG, the game manager has to set over than a hundred different parameters for the simplest scenario (one region, one product, two bank accounts) (see Figure 10). To help perform this task special stand-alone application GameEditorPro was developed (see Figure 11). It generates a full set of the game scenario parameters according to guidelines provided by the game manager.

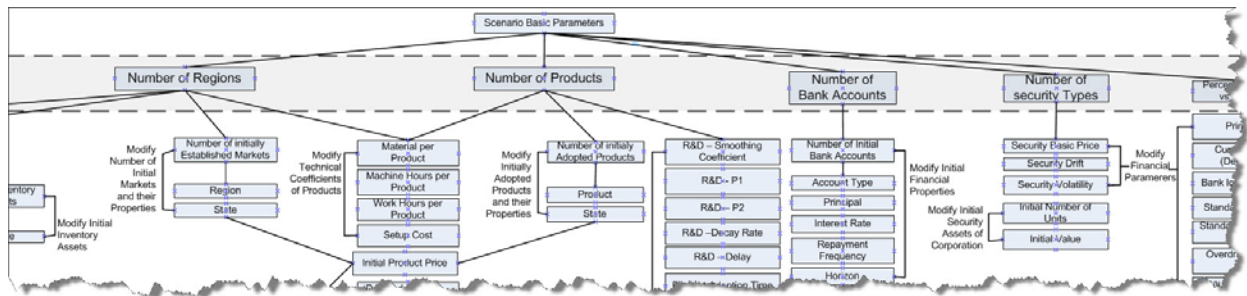


Figure 10: Examples of the ILMG parameters

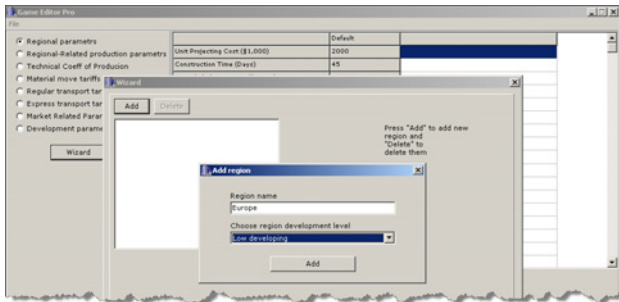


Figure 11: GameEditorPro Software Interface

The game manager decides only about number of regions and products as well as their characteristics; then the software generates all other parameters automatically. Regions have three possible characteristics: with (1) advanced, (2) less advanced and (3) developing economy. Products can be (1) hi-tech, (2) "medium"-tech and (3) low-tech. The following algorithm is proposed for the automatic generation of parameters:

- the full range of parameters' values is defined;
- each of defined range is divided in three equal subsets;
- each of parameters' subset characterizes either one of region type: (1) advanced, (2) less advanced and (3) developing economy, or product type: be (1) hi-tech, (2) "medium"-tech and (3) low-tech;
- within each subset values of parameters are generated randomly.

Further, the game is used to simulate generated scenario and to display the results. Here the procedure of dynamic scenario update is not yet explored.

7. CONCLUSIONS

An agent-directed multisimulation framework for scenario simulation in management simulation games is proposed in the paper. Three possible roles for agents in games are considered: scenario manager, virtual player, and virtual tutor. Though, the use of agents in simulation in general and in simulation games in particular is widespread, however combination of multisimulation with agent technology is a novel concept that seems to be very promising for use in simulation games, especially for scenario management in interacting, computer-aided and internet-based, real-time processed business

simulations. The example of such game is considered in the final chapter of the paper.

REFERENCES

- Bai X., Black J.B., Vitale J., 2007. Learning with the Assistance of a Reflective Agent. URL: http://www.ilt.columbia.edu/projects/REAL_ABSHL2007_final.doc
- Bikovska J., Merkurjeva G., Grubbström R.W., 2006. Enhancing Intelligence of Business Simulation Games. *20th European Conference on Modelling and Simulation*. Proceedings of the International Conference, May 28-31, 2006, Bonn, Sankt Augustin, Germany, p.641-646.
- Capuano N., De Santo M., Marsella M., Molinara M., Salerno S., 2000. A Multi-Agent Architecture for Intelligent Tutoring. URL: http://www.capuano.biz/Papers/SSGRR_2000_P1.pdf
- Dobson Mike, Kyrilov Vadim, Kyrylova Tetyana, 2004. Decision Training Using Agent-Based Business Strategy Games. *Proceedings of the Seventh IASTED International Conference: Computers and advanced Technology in Education*. August 16-18, 2004, Kauai, Hawaii, USA. Pages 66-71.
- Grubbström R.W., Merkurjeva G., Bikovska J., Weber J. 2005. "ILMG: Learning Arrangements and Simulation Scenarios." In *Proceedings of 19th European Conference on Modelling and Simulation*. Riga, Latvia, June 1-4, p. 715-720.
- Grubbström, R.W., Bikovska, J., *International Logistics Management Game. Participants' Manual*, Linköping 2003.
- Hospers M., Kroezen E., Nijholt A., Op Den Akker R., Heylen D., 2003. An Agent-based Intelligent Tutoring System for Nurse Education. URL: http://wwwhome.cs.utwente.nl/~anijholt/artikelen/ctit18_2003.pdf
- Ören, T.I., *Model Update: A Model Specification Formalism with a Generalized View of Discontinuity*, (Proceedings of the Summer Computer Simulation Conference, Montreal, Quebec, Canada, 1987 July 27-30), pp. 689-694.
- Ören, T.I., *Dynamic Templates and Semantic Rules for Simulation Advisors and Certifiers*, In: P.A. Fishwick and R.B. Modjeski (eds). *Knowledge-Based Simulation: Methodology and Application*, (Springer-Verlag, New York, 1991), pp. 53-76.
- Ören, T.I. (2000 - Invited Paper). Agent-directed Simulation - Challenges to meet Defense and Civilian Requirements. Proc. of the 2000 Winter Simulation Conference, J.A. Joines et al., eds., Dec. 10-13, 2000, Orlando, Florida, pp. 1757-1762.
- Ören T., 2001. 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. and F. Longo (2008). Emergence, Anticipation and Multisimulation: Bases for Conflict Simulation. Proceedings of the EMSS 2008- 20th European Modeling and Simulation Symposium. (Part of the IMMM-5 - The 5th International Mediterranean and Latin American Modeling Multiconference), September 14-17, 2008, Campora San Giovanni, Italy, pp. 546-555.
- Remondino M., 2007. An Overview of Agent Based Paradigms and Enterprise Simulation for E-Learning and Knowledge Transmission. *Proceedings of the I3M2007 Conference*. October 4-6, 2007. Bergegg, Italy.
- Remondino M., 2008. A Web Based Business Game Built on System Dynamics Using Cognitive Agents as Virtual Tutors. Proceedings of the Tenth International Conference on Computer Modeling and Simulation (uksim 2008). Pages 568-572.
- Yilmaz, L. (ed.) (2005). Agent-Directed Simulation. SCS, San Diego, CA
- 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., Ören T., 2004. Dynamic Model Updating in Simulation with Multimodels: A Taxonomy and a Generic Agent-Based Architecture, Proceedings of SCSC 2004 - Summer Computer Simulation Conference, July 25-29, 2004, San Jose, CA., pp. 3-8.
- Yilmaz L., Ören T., 2006. Intelligent agents, simulation, and gaming. *Simulation & Gaming, Vol 37 No. 3, September 2006, p. 339-349*.
- Yilmaz, L. and T.I. Ören (eds.) (2009-in Press)). Agent-Directed Simulation and Systems Engineering. Wiley Series in Systems Engineering and Management, Wiley-Berlin, Germany.
- Yilmaz L., Ören T.I., Ghasem-Aghaee N., 2006 . Simulation-based problem-solving environment for conflict studies: Toward Exploratory Multisimulation with Dynamic Simulation Updating. *Simulation & Gaming, Vol 37 No. 4, September 2006, p. 534-556*. DOI (Digital Object Identifier): 10.1177/1046878106292537.

Van Luin J., Tulba F., Wagner G., 2004. Remodeling the Beer Game as an Agent-Object-Relationship Simulation, *Proceedings of Workshop 2004: Agent-Based Simulation 5*, Lisbon, Portugal, 3-5 May 2004, SCS European Publishing House, 2004.

AUTHORS BIOGRAPHIES

GALINA MERKURYEVA is Professor at the Department of Modelling and Simulation at Riga Technical University, Latvia. Her research interests are in the fields of discrete-event simulation, simulation metamodeling and optimisation, simulation-based training, supply chain simulation and decision support.

JANA BIKOVSKA is PhD student and teaching assistant at the Department of Modelling and Simulation, Riga Technical University, Latvia. Her professional interests are in the field of supply chain management, business simulation games, scenario approach in complex systems with application to simulation games.

TUNCER ÖREN is a professor emeritus of Computer Science at the University of Ottawa. His current research activities include advanced M&S methodologies such as multimodels, multisimulation and emergence; agent-directed simulation; cognitive simulation; and 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 received "Information Age Award" from the Turkish Ministry of Culture, Distinguished Service Award from SCS and plaques and certificates of appreciation from organizations including ACM, AECL, AFCEA, and NATO.

THE EFFECTS OF BRAND EQUITY ON PRICE STRATEGIES: AN AGENT BASED MODEL

Sebastiano A. Delre^(a), Skander Essegaier^(b)

^(a) Management Department
Bocconi University
Via Roentgen 1, 20136 Milan, Italy

^(b) Marketing Department
Koç University
College of Administrative Sciences and Economics, CAS 157, Istanbul, Turkey

^(a) [Email: sebastiano.delre@unibocconi.it](mailto:sebastiano.delre@unibocconi.it) ^(b) [Email: sesseghaier@ku.edu.tr](mailto:sesseghaier@ku.edu.tr)

ABSTRACT

Consumers are highly sensible to different price structures and price promotions. Many studies have showed how customers differently respond when prices are split into separate parts, e.g. a regular price and a shipping and handling surcharge. This phenomenon has recently received much more attention because online sales have continuously and substantially increased in the last years and because online sales imply a price partitioning: product price and shipping price. This gives opportunities to online retailers. They can decide whether to apply promotional tactics on both regular prices and on shipping and handling prices. The price partitioning decision becomes more complicated than usual. Retailers have to choose between a free shipping offer strategy and a price partitioning strategy. In the former case they have to decide a single price that includes the shipping cost and in the latter cases they have to choose upon two prices, a price for the item and a price for the shipping. This paper investigates how firms decide which of these two strategies to adopt and how their brand equities affect their decisions. An agent based model is built in order to replicate Gümüş et al. (2009) results and to depart from it bringing new insights about market partitioning (how many firms adopt which strategy) and the effects of brand equities.

Keywords: price partitioning, brand equity, agent based models

1. INTRODUCTION

Consumers are highly sensible to different price structures and price promotions. Many studies have showed how customers differently respond when prices are split into separate parts, e.g. a regular price and a shipping and handling surcharge (Campanelli 2002, Chakravarthi 1988, Dinlersoz and Li 2004, Gomolski 2001, Schindler et al. 2005, Smith and Brynjolfsson 2001). This phenomenon has recently received much

more attention because online sales have continuously and substantially increased and because online sales often imply a price partitioning: product price and shipping price.

This gives opportunities to online retailers. They can decide whether to apply promotional tactics on both regular prices and on shipping and handling prices (Yang et al. 2008). Retailers have to choose between a free shipping offer strategy and a price partitioning strategy. In the former case they have to decide a single price that includes the shipping cost and in the latter cases they have to choose upon two prices, a price for the item and a price for the shipping.

This paper investigates how firms decide which of these two strategies to adopt and how their brand equities affect their decisions. An agent based model is built in order to replicate Gümüş et al. (2009) results and in order to depart from it bringing new insights about market partitioning (how many firms adopt which strategy) and about the effects of brand equities.

2. THE MODEL

Gümüş et al. (2009) model consists of a two-stage game. First, firms decide which strategy to use between a zero-cost shipping strategy (ZS strategy) and a price partitioning strategy (PS strategy). Then, they decide what prices to charge. Demand is divided into two segments: skeptics and non-skeptics. While skeptics are assumed to buy only from firms that use ZS strategy, non-skeptics buy from any firm. This is a key assumption of the model because it generates a crucial trade off: on one hand the ZS strategy encounters higher costs but it guarantees a bigger demand because it includes both skeptics and non-skeptics; on the other hand the PS strategy may obtain higher margins because it avoids the costs of shipping but it loses a segment of the demand because skeptics do not buy from firms adopting it.

We build the following agent based model in order to replicate and extend Gümüş et al. (2009) results. There exist N firms deciding at each time step whether to adopt a ZS strategy or a PS strategy. At each time step firm i computes expected demand for the ZS strategy (1), expected demand for the PS strategy (2), expected profit for the ZS strategy (3) and expected profit for the PS strategy (4). Then it chooses the strategy with the highest expected profit. In case the expected profits are the same, firm i uses the same strategy it used at the previous time step (7). Both expected demands consist of two parts: the market potential and the price effect. Market potential for the ZS strategy consists of both skeptics (α) and non-skeptics ($1-\alpha$) and market potential for the PS strategy consists of only non-skeptics ($1-\alpha$). Here γ_i represents the brand equity of firm i . The price effect is the same for both demands, it is assumed to have an exponential form that decreases in its own price (depending on a) and in all the others' firm prices (depending on b). Concerning the profits, for the ZS strategy the profit consists of the demand times the margin (price minus costs) and for the PS strategy the profit consists of the same as the ZS strategy plus an extra percentage in order to cover shipping fees (this is formalized by s). Here costs are assumed to decrease with brand equity in order to formalize economics of scale and/or scope. When all firms are assumed to have similar popularity (the same brand equity), costs become a constant (5); when brand equity varies in the population of firms, costs are assumed to decrease linearly respect to the brand equity (6).

$$d_{it}^{ZS} = \left[\frac{\gamma_i}{\sum_{k=1}^{N_{ZS}} \gamma_k} \cdot \alpha + \gamma_i \cdot (1-\alpha) \right] \cdot \exp \left(-a \cdot p_i + \sum_{j=1}^{N_{\{-i\}}} b \cdot p_j \right) \quad (1)$$

$$d_{it}^{PS} = [\gamma_i \cdot (1-\alpha)] \cdot \exp \left(-a \cdot p_i + \sum_{j=1}^{N_{\{-i\}}} b \cdot p_j \right) \quad (2)$$

$$\pi_{it}^{ZS} = d_{it}^{ZS} \cdot [p_i^{ZS} - C] \quad (3)$$

$$\pi_{it}^{PS} = d_{it}^{PS} \cdot [(1+s) \cdot p_i^{PS} - C] \quad (4)$$

$$C = c \quad (5)$$

$$C(\gamma_i) = m - \gamma_i \quad (6)$$

$$F_{i,t+1} = \begin{cases} \pi_{it}^{PS} < \pi_{it}^{ZS} & \rightarrow ZS \\ \pi_{it}^{PS} > \pi_{it}^{ZS} & \rightarrow PS \\ \pi_{it}^{PS} = \pi_{it}^{ZS} & \rightarrow F_{i,t} \end{cases} \quad (7)$$

3. THE MODEL

Gümüş et al. (2009) solve the game by backward induction. First optimum prices are calculated (p^{ZS} and p^{PS}). Then, given those optimum prices, equilibrium proportions of firms that adopt each of the two strategies are derived. Their main contribution consists of the following two main results:

- (a) Market partitioning: how the market is structured, i.e. the proportion of firms adopting each of the two strategies;
- (b) Brand equity effect: how strategy adoption depends on the distribution of brand equities.

However, these results strongly depend on the stringent assumptions they make in order to solve the game analytically. In particular they find the proportions of firms adopting each strategy when firm brands are assumed to be equal and they find how the market equilibrium will change for some brand equities when there are only two firms in the market.

3.1. Market partitioning

The first result of Gümüş et al. (2009) consists of finding how many firms adopt the ZS strategy and how many adopt the PS strategy. Assuming that firms are equally popular and that consequently costs are constant (5), Gümüş et al. (2009) found a unique pure strategy Nash equilibrium (8).

$$\beta^* = \frac{N_{ZS}^*}{N} = \min \left(\frac{\frac{\alpha}{1-\alpha}}{(1+s) \cdot \exp \left(\frac{acs}{(1+s)} \right) - 1}, 1 \right) \quad (8)$$

Our agent based model consistently replicates this result. We set up the agent based model with the following parameters' values: $N=100$; $a=1$; $b=1$; $c=\{0.5, 1, 2, 3, 5, 10\}$; $\alpha=\{0.1, 0.3, 0.5, 0.7, 0.9\}$; $s=\{0.1, 0.3, 0.5, 0.7, 0.9\}$; $p^{ZS}=c+1/a$; $p^{PS}=c*s/(1+s)+s/a$ where p^{ZS} and p^{PS} are the optimum equilibrium prices given a firm decides to opt for the ZS strategy and the for PS strategy, respectively. Simulations run 50 times for each condition guarantying convergence of the average outputs. Under all possible combinations of these parameters' values we always obtained that the proportion of firms adopting the ZS strategy converged towards the value of β^* .

We depart from this result relaxing the assumption of equal popularity and constant costs. We adopted (6) instead of (5) and we let γ_i values to be randomly assigned according to a beta distribution. In this way we are able to constrain γ_i between 0 and 1 and to manipulate the mean and the standard deviation of its distribution. Other parameters of the model are set to the following values: $N=100$; $a=1$; $b=1$; $\alpha=0.1$; $s=0.5$; $m=1$. Figure1 reports results of 45 different simulation conditions varying the mean and the standard deviation

of the distribution of γ_i . For each condition the simulation is repeated 50 times and the averages are reported in the graph. It is evident that for higher values of γ_i , the proportion of firms adopting the ZS strategy substantially decreases. This indicates that when firms are more popular and obtain lower costs, the PS strategy tend to become the most efficient. This result seems quite realistic: on the one hand, in an immature market, characterized by unknown brands, firms tend to offer a free shipping service trying, in this way, to gain market potential and popularity. On the other hand, in a mature market with consolidated strong brands and where the demand is presumably well exploited, firms tend to opt for price partitioning in order to increase their margins.

Moreover the graph indicates that also the differentiation of brand equities affects the structure of the market. For very low values of γ_i , if there are differences in the market in terms of brand popularity, firms tend to adopt more the PS strategy. This relation is inverted when firms have higher γ_i values. For immature markets, if some firms happen to be endorsed with more popularity, they tend to opt for the PS strategy in order to obtain higher margins. Consequently they are copied by those firms that have similar popularity. Contrarily, in very mature market this tendency is reversed because for very well known firms it makes no sense anymore to compete against other strong brands on a single segment (skeptics). Because many firms have high and similar popularity, it makes more sense to compete on the whole market (both skeptics and non-skeptics), thus adopting a ZS strategy.

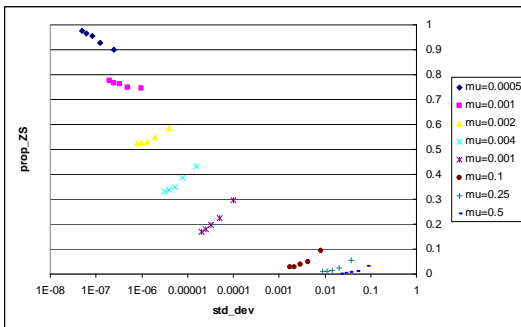


Figure 1. Proportion of Firms Adopting ZS strategy Varying the Distribution of Brand Equities (Different Means and Different Standard Deviations).

3.2. Brand equity effect

Gümüş et al. (2009) are able to exploit the relation between brand equity and strategy selection by simplifying the game reducing the number of firms to 2 (where, for simplicity it is assumed that $\gamma_1 \geq \gamma_2$ and $\gamma_1 = 1 - \gamma_2$). In this case Gümüş et al. (2009) find that there exists a given threshold γ^* for which the followings hold:

- If both firms have a brand equity greater than γ^* , i.e. $\min(\gamma_1, \gamma_2) \geq \gamma^*$, then both firms opt for the ZS strategies.
- If only one firm has a brand equity greater than γ^* , i.e. $\gamma_1 \geq \gamma^*$ and $\gamma_2 \leq \gamma^*$, then firm1 (F1) opts for the ZS strategy and firm2 (F2) opts for the PS strategy.

The value of γ^* is given by equation (9):

$$\gamma^* = m + \frac{1+s}{a \cdot s} \cdot \log[(1+s) + (1+\alpha)] \quad (9)$$

Gümüş et al. (2009) do not specify what happens when both firms have brand equities lower than γ^* , i.e. $\max(\gamma_1, \gamma_2) < \gamma^*$.

We set up the agent based model with the following parameters' values: $N=2$; $a=1$; $b=1$; $m=1$; $p^{ZS}=c+1/a$; $p^{PS}=c \cdot s / (1+s) + s/a$; $\alpha = \{0.05, 0.1, 0.15, \dots, 0.9, 0.95, 1\}$ (20 values); $s = \{0.05, 0.1, 0.15, \dots, 0.9, 0.95, 1\}$ (20 values); $\gamma_1 = \{0.05, 0.1, 0.15, \dots, 0.9, 0.95, 1\}$ (20 values) and $\gamma_1 = 1 - \gamma_2$. Consequently we obtained 8000 simulation conditions. For every condition for which $\min(\gamma_1, \gamma_2) \geq \gamma^*$, or $\gamma_1 \geq \gamma^*$ and $\gamma_2 \leq \gamma^*$, we always find that F1 and F2 behave accordingly to what predicted by Gümüş et al. (2009). This represents a complete replication of Gümüş et al. (2009) second result.

We depart from this result exploring what happens at simulation conditions for which both firms have brand equities lower than γ^* , i.e. $\max(\gamma_1, \gamma_2) < \gamma^*$. As Table 1 illustrates, we obtain a total of 2736 simulation conditions where this holds. We create 2 variables, a dummy variable indicating whether the simulation converges or not and a nominal variable which indicates which one of the 4 possible strategy selections we obtain at the end of the simulation run. We find that there is never a convergence towards F1=ZS, F2=ZS. This means that F1=ZS, F2=ZS is never a steady state equilibrium point. Moreover we find that F1=ZS, F2=PS; F1=PS, F2=ZS and F1=PS, F2=PS can be steady state equilibrium points. This means that there are parameters' values for which there is convergence towards them. It happens more often for F1=PS, F2=PS, less often for F1=ZS, F2=PS and for F1=PS, F2=ZS. Finally, when there is not convergence towards a steady state equilibrium point the only possible outcome we observe is a 2-cycle equilibrium, i.e. firms continuously jump from F1=ZS, F2=ZS to F1=PS, F2=PS. These results indicate that for very immature markets, in which it is more likely that all firms have similar brand popularities and these are below a given threshold, the strategy selection is much more unpredictable than more mature markets.

The simulation experiment specified above allows us to study the effects of the parameters on the convergence of the simulation output and on the strategy selection. We run a binary logistic regression in order to measure the effects of γ_1 , (γ_2 is simply a

function of γ_1 , s and α on the convergence and on the strategy selection. Table2 presents the results. We find that γ_1 does not significantly determine convergence, s does it positively and α does it negatively.

Concerning strategy selection we find that the 3 parameters always affect strategy selection for any of the 4 possible cases, except for γ_1 not affecting F1=ZS, F2=ZS. It is interesting to notice the opposite effects of s and α indicating the fact that in order to efficiently predict any strategy selection outcome, if there are many skeptics (α is high) it is necessary that shipping fees are low (s is low) and if there are a few skeptics (α is low) it is necessary that shipping fees are high (s is high).

Table1. Crosstab between Convergence and All Possible Strategy Selections.

	Strategies			
	F1=zs and F2=zs	F1=zs and F2=ps	F1=ps and F2=zs	F1=ps and F2=ps
No converg.	125	0	0	108
Converg.	0	603	722	1178

Table2. The effect of γ_1 , s , and α on the convergence of the output and on the strategy selection.

	Beta	S.E.	Wald	df	Sig.
Convergence					
gamma1	-.003	.344	.000	1	.993
s	11.517	.821	196.599	1	<.001
alpha	-26.487	1.640	260.775	1	<.001
F1=zs and F2=zs					
gamma1	.175	.437	.159	1	.690
s	-9.222	.908	103.208	1	<.001
alpha	21.898	1.775	152.239	1	<.001
F1=zs and F2=ps					
gamma1	-9.166	.418	480.653	1	<.001
s	-3.685	.320	132.566	1	<.001
alpha	10.561	.632	278.936	1	<.001
F1=ps and F2=zs					
gamma1	9.462	.400	559.163	1	<.001
s	-3.372	.301	125.412	1	<.001
alpha	9.805	.597	269.746	1	<.001
F1=ps and F2=ps					
gamma1	-.842	.166	25.676	1	<.001
s	3.811	.222	293.434	1	<.001
alpha	-12.287	.493	621.233	1	<.001

4. CONCLUSION AND FUTURE RESEARCH

The main new results of this work consist of predicting firms price strategies for market structures where brand equities are distributed in different ways. Future work could further extend the simulation model relaxing some strong assumptions. For example we could opt for a more realistic non linear cost function and check whether our results still hold. Moreover we could introduce a different segmentation of the market where the demand includes also a third segment of consumers that differently weight regular prices and shipping and handling surcharges. In this way we could explore how the structure of the market could respond to different kind of demands.

Finally a possible development of this work, in order to strengthen and corroborate its implications could incorporate an empirical validation of its simulation results. In this way the contribution of the work could address precisely markets where the characteristics that our model takes into considerations are relevant and others markets where they are not (Fagiolo et al 2007, Wildrum et al. 2007, Janssen and Ostrom, 2006).

REFERENCES

- Campanelli M (2002). Who pays to get it there? Net profits: your site's shipping and handling charges can make or break an online sale, *Entrepreneur*, 30, (2), 34-36.
- Chakravarthi N (1988). Competitive promotional strategies, *Journal of Business*, 61, (4), 427-449.
- Dinlersoz E and Li H (2004). Shipping news: an analysis of internet retailers' shipping strategies, working paper, University of Huston.
- Gomolski B (2001). E-business matters: shipping and handling costs leave a bad taste on the lips of e-shoppers, *InfoWorld*, 23, (4), 72.
- Gümüş M, Li S, Oh W and Ray S (2009). Shipping fees or shipping free? Impact of product and retailer characteristics on shipping charges in e-commerce, working paper.
- Fagiolo G, Birchenhall C and Windrum P (2007). Empirical validation in agent-based Models: Introduction to the Special Issue, *Computational Economics*, 30, 189-194.
- Janssen MA, Ostrom E (2006). Empirically based, Agent-based Models, *Ecology and Societies*, 11, (2), 37.
- Windrum P, Fagiolo G and Moneta A (2007). Empirical Validation of Agent-Based Models: Alternatives and Prospects, *Journal of Artificial Societies and Social Simulation*, 10, (2).
- Schindler RM, Morrin M and Bechwati NN (2005). Shipping charges and shipping-charges skepticism: implications for direct marketers' pricing formats, *Journal of Interactive Marketing*, 19, (1), 41-53.
- Smith M and Brynjolfsson E (2001). Consumer decision-making at an internet shopbot: brand still

matters, *Journal of Industrial Economics*, 49, 541-558.

Yang Y, Essegai S and Bell DR (2008) The fixed costs of shopping on the internet, working paper, The Wharton School, University of Pennsylvania.

CONSIDERING WORKPIECES AS INTEGRAL PARTS OF A DEVS MODEL

Christina Deatcu^(a), Thorsten Pawletta^(b), Olaf Hagendorf^(c), Bernhard Lampe^(d)

^(a,b)Hochschule Wismar, University of Applied Sciences: Technology, Business and Design, Germany

^(c)Liverpool John Moores University, School of Engineering, UK

^(d)University of Rostock, Germany

^(a)christina.deatcu@hs-wismar.de, ^(b)thorsten.pawletta@hs-wismar.de,
^(c)oh@ibhagendorf.de, ^(d)bernhard.lampe@uni-rostock.de

ABSTRACT

In the area of modelling and simulation of production systems often the flow of workpieces is subject of study. Workpieces or, more generalised entities, are sent through a system and are modified by passive elements of the system. In modular, hierarchical models, from the system theoretic point of view, they are usually not seen as components of the system, but just as data structures passed through it. That means that dynamic behaviour of the entities themselves is modelled indirectly. For modular, hierarchical DEVS systems the alternative of modelling the workpieces as atomic systems and considering them as integral parts of the system is proposed. Especially for hybrid DEVS systems this modelling approach offers several advantages. Using the example of workpieces being heated in an oven, required data structures and algorithms for dynamic structure hybrid DEVS are presented.

Keywords: DEVS, hybrid modelling, dynamic structure, scientific and technical computing environments

1. INTRODUCTION

The Discrete Event System Specification (DEVS) formalism introduced by Zeigler (Zeigler 1976) offers good opportunities to describe real world systems in the engineering area as e.g. complex production lines. The modular, hierarchical construction of models for production lines thereby should result in models, which reflect the reality in a 1-to-1 manner. This means, an engineer creating such a model or simulating it should recognise elements and their behaviour from the real plant, as e.g. machines.

For simulation, typically entities representing the workpieces are passed through the system. These workpieces are usually not regarded as integral parts of the model. The system structure remains static during simulation and entities representing the workpieces are routed through the system causing events when arriving at model components. Therefore, moving entities can be seen as the engine triggering the progress of the simulation.

Workpieces can have a dynamic behaviour themselves, often of continuous type. The first thing to happen if continuous parts appear in a DEVS model, is that classic DEVS needs to be extended to hybrid DEVS. Definitions for hybrid atomic DEVS where evolved by Praehofer (Praehofer 1992). If those continuous parts are the workpieces, hybrid DEVS already offers opportunities to model such kind of systems. The dynamic behaviour of the workpieces is thereby represented indirectly by specifying the differential equations in model components where entities pass by.

To reflect real world structure better, it is obvious to model dynamic behaviour where it is really located, i.e. in the workpiece itself. Following this approach, the workpieces need to be seen as full-fledged dynamic system components and therefore have to be into the system model. Since the integration of workpieces into the model causes model structure changes during simulation runtime, the resulting DEVS model has to be of dynamic structure.

Approaches which extend the classic DEVS to dynamic structure DEVS (DSDEVS) add functionality to the coupled model, while the definition of atomic DEVS remains unchanged (Barros 1996; Pawletta, Lampe, Pawletta, and Drewelow 2002). Another approach are agent based concepts, introduced by Uhrmacher and Arnold (Uhrmacher and Arnold 1994) and others. The combination of several DEVS extensions as DSDEVS, Parallel DEVS (PDEVs) and DEVS with ports is proposed by Hagendorf et al. (Hagendorf, Pawletta and Deatcu 2009).

The combined hybrid and dynamic structure approach following Pawletta et al. (Pawletta, Lampe, Pawletta, and Drewelow 2002; Pawletta, Deatcu, Hagendorf, Pawletta and Colquhoun 2006) is suitable for the representation of workpieces with continuous dynamics as integral parts of a DEVS model.

In the following, after a short overview on DEVS, fundamental aspects of modelling moving entities, as e.g. workpieces in a production line, with integrated dynamic behaviour are investigated. At first, the classic pure data structure based approach is examined. After that, the employment of a dynamic structure modelling

approach is discussed. Differences in modelling workpieces with own dynamic behaviour are described and advantages of integrating them into the system are depicted.

2. A BRIEF REVIEW OF DEVS

Modular, hierarchical DEVS models are composed of two model types, atomic and coupled models. An overview of the formalisms that underpin and extend DEVS theory is given by Zeigler et al. (Zeigler, Kim, and Praehofer 2000).

Since it is aimed to model continuous dynamic behaviour of workpieces, the need to benefit from hybrid DEVS approaches is obvious. Hybrid means that beside discrete model fractions, continuous model parts are contained, as well. As an extensions of the basic DEVS formalism the hybrid DEVS formalism presented by Praehofer (Praehofer 1992) is employed, which facilitates integration of existing ODE solvers. Another approach for hybrid DEVS modelling was introduced by Kofman (Kofman 2004; Cellier and Kofman 2006), who proposes quantisation of the state variables instead of time discretisation to approximate differential equations.

2.1. Atomic Model Specification

Praehofer defines hybrid characteristics on the atomic system level as follows:

$$A_{\text{hybrid}} = (X, Y, S, f, c_{se}, \lambda_c, \delta_{x\&s}, \delta_{int}, \lambda_d, ta) \quad (1)$$

where X , Y , and S specify the set of inputs, outputs and states which may be continuous or discrete. Continuous dynamics are mapped by the rate of change function f and the continuous output function λ_c . State event conditions of continuous quantities are defined in c_{se} and state events, beside external events, induce state transitions using the function $\delta_{x\&s}$. Internal events activate the discrete output function λ_d and also the internal state transition function δ_{int} . After each discrete state transition internal events are rescheduled by the time advance function ta . Table 1 gives an overview on hybrid atomic DEVS sets and characteristic functions.

Table 1: Elements of Hybrid Atomic DEVS

X	set of input values
Y	set of output values
S	set of states
f	rate of change function
c_{se}	state event condition function
λ_c	continuous output function
$\delta_{x\&s}$	external and state event transition function
δ_{int}	internal transition function
λ_d	discrete output function
ta	time advance function

2.2. Classic Coupled Model Specification

Coupled systems describe a static aggregation of atomic and/or coupled systems, and they permit the construction of hierarchical structures. If couplings are restricted to equivalence relations, a coupled system consists of a set of subsystems and couplings. A coupled system is defined (Praehofer 1992) by the tuple:

$$N = (X_N, Y_N, D, \{M_d \mid d \in D\}, EIC, EOC, IC, Select) \quad (2)$$

where X_N is the set of input values and Y_N is the set of output values. The index N stands for network model as synonym for coupled model. D is the set of the component names, while M_d represents a dynamic subsystem. The property of ‘closure under coupling’, which is ensured for classic coupled DEVS, enables the representation of every coupled DEVS as an atomic DEVS. Thus, dynamic subsystems may be other coupled or atomic DEVS models. EIC , EOC , and IC define the coupling relations. The external input coupling relation EIC connects external inputs to component inputs, the external output coupling EOC connects component outputs to external outputs and the internal coupling IC defines connections among components, i.e. component outputs are connected to component inputs. For couplings no direct feedback loops are allowed. Finally, $Select$ acts as a special function to prioritise one subsystem in case of simultaneous internal events in subsystems. Table 2 subsumes coupled DEVS elements.

Table 2: Elements of Coupled DEVS

X_N	set of input values
Y_N	set of output values
D	index set of M_d
M_d	a dynamic subsystem
EIC	external input coupling
EOC	external output coupling
IC	internal coupling
$Select$	tie-breaking function

3. REPRESENTATION OF WORKPIECES

The example of a section of a production line is used to point out the different modelling strategies in detail and to discuss pros and cons of the approaches. It was developed in dependence on and extension of the well known sample problem of a soaking pit furnace (Ashour and Bindingnavle 1973). The analysed section of the system is rather small, but adequate to illustrate the fundamental concept of considering workpieces as integral parts of a DEVS model.

An oven, shown in figure 1, with the ability to hold up to three workpieces at a time, which is part of the production line with multiple heating facilities, is modelled.

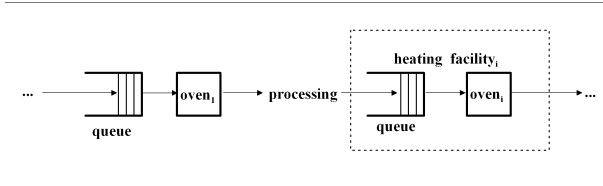


Figure 1: Section of Production Line

Temperature in the oven interacts with temperature of the workpieces being heated. The heat dissipation of workpieces during waiting time in the queue in front of the oven can be taken into account, too. The model contains both, discrete and continuous, parts. Hybrid atomic DEVS models are employed to map the continuous heating process and the cooling-down of workpieces, i.e. they are used to model the march of temperature.

3.1. Workpiece Modelled as Data Structure

Usually workpieces are modelled as data structures without own dynamic behaviour. In a simulation they are used as moving entities triggering events in dynamic system components. However, in the example described workpieces hold an own dynamic behaviour. It is characterised by their temperature, which is saved in a workpiece related state variable \mathcal{G} . The continuous temperature behaviour of a workpiece can be specified by an ordinary differential equation of the following type:

$$\dot{\mathcal{G}} = f(\dot{Q}_{in}, \dot{Q}_{out}, \mathcal{G}, \dots) \quad (3)$$

If we investigate workpieces consisting of different material compositions (composite materials), we need to specify a particular differential equation for each workpiece type. Simple parameterisation of the equations is not possible, since they are of different structure.

Due to the fact that in traditional modelling approaches moving entities cannot specify an own dynamic behaviour, the differential equations, required to calculate the dynamics of workpieces' temperature, need to be defined in all atomic systems where workpieces pass by. That means dynamic behaviour of the workpieces is modelled indirectly in atomic model parts. If the data structure of a workpiece arrives at an atomic model component, this component needs to perform a case differentiation to check out which specific differential equation has to be applied. This case differentiation is based on the parameter id of the workpiece, which describes the workpiece type. Current workpiece temperature \mathcal{G} is passed from the workpiece to the atomic model, where temperature change is calculated until the workpiece leaves the system after receiving the updated value for \mathcal{G} .

Figure 2 presents the fundamental model structure of a heating facility specified as a classic coupled DEVS system with two hybrid atomic DEVS systems according to the formal definitions in (1) and (2).

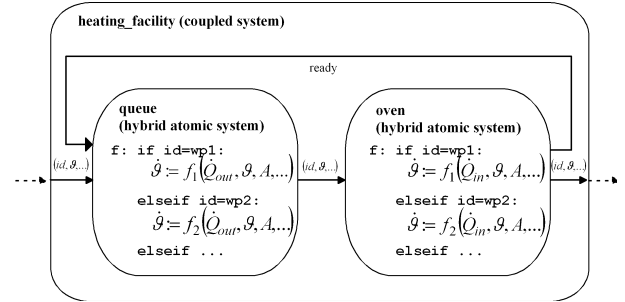


Figure 2: Modular, Hierarchical DEVS Model

The implementation of the rate of change functions f of the hybrid atomic systems *queue* and *oven* is adumbrated to demonstrate that maintenance and enlargement of such a model is rather elaborate, complicated and time consuming. If e. g. the model should be qualified to deal with another type of workpiece, the rate of change function in all concerned atomic components needs to be adopted.

3.2. Workpiece Modelled as an Atomic System

To avoid the modeller from having to adopt the dynamic descriptions distributed over the whole model, it seems to be obvious to concentrate the dynamic specification where it is really located, namely in the workpiece itself. Additionally, more autonomous workpieces become imagineable. For example, depending on the temperature already reached, workpieces could request individually to leave the oven.

Figure 3 shows a graphic representation of the atomic model for one type of workpiece.

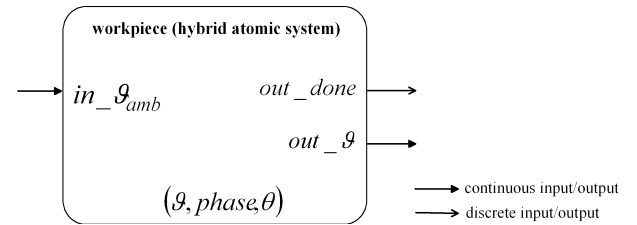


Figure 3: Atomic System Workpiece

Besides the states \mathcal{G} , $phase$ and σ the atomic model keeps a specific parameter vector of system parameters θ that is defined as following:

$$\theta = (k, A, m_{wp}, c_{wp}, \mathcal{G}_{target}) \quad (4)$$

The workpiece model takes the ambient temperature as input and propagates its own temperature. Furthermore, a discrete output for a workpiece-done event is established. Formal description of the atomic model *workpiece* is as follows:

$$\begin{aligned} workpiece_{hybrid} &= (X, Y, S, f, c_{se}, \lambda_c, \delta_{x\&s}, \delta_{int}, \lambda_d, ta) \quad (5) \\ X &= \{ in_g_{amb} \} \\ Y &= \{ out_g, out_done \} \end{aligned}$$

$$S = \left\{ (\vartheta, phase, \sigma) \mid \vartheta \in \mathfrak{R}^+, phase \in \{ unfinished, done \}, \dots, \sigma \in \mathfrak{R}^+ \right\}$$

with characteristic functions:

$$f(\vartheta, \theta, \vartheta_{amb}) = -\frac{k \cdot A}{m_{wp} \cdot c_{wp}} \cdot (\vartheta - \vartheta_{amb}) = (\dot{\vartheta})$$

$$c_{se}(\vartheta, \theta) = \begin{cases} true & \text{if } \vartheta \geq \vartheta_{target} \\ false & \text{otherwise} \end{cases}$$

$$\lambda_c(\vartheta) = (\vartheta)$$

$$\delta_{x\&s}(X, e, \sigma) = (\vartheta, phase, \sigma)$$

% no discrete external events
% state event:
phase = done % change state
σ = 0 % activate internal event

$$\delta_{int}(\vartheta, phase, \sigma) = (\vartheta, phase, \sigma)$$

phase = unfinished % reset state
σ = inf % deactivate internal event

$$\lambda_d(phase) = (phase) \text{ % sends done-signal}$$

$$ta(\sigma) = (\sigma)$$

where σ is a variable to store the time until next internal event.

If the workpieces are modelled as atomic systems which ‘travel’ through the system instead of as simple data structures, a totally different model architecture is required. Furthermore, the modelling formalism needs to provide methods to handle structure changes on coupled systems level. Required data and model structure as well as relevant events for modelling and simulation are described in chapter 4.

4. DYNAMIC STRUCTURE MODELLING

If the different types of workpieces are specified as atomic systems and workpieces are considered to be temporary atomic systems which ‘travel’ through the system during simulation runtime, an extended definition of coupled DEVS systems is required. The dynamic structure DEVS (DSDEVS) approach delivers appropriate extensions, where DSDEVS allows to create, delete and move the atomic systems representing the workpieces from coupled model to coupled model.

4.1. Dynamic Structure Hybrid DEVS

A dynamic structure system in DEVS context is a modular, hierarchical system whose structure changes during simulation runtime. The dynamic behaviour of a system is reflected in atomic models, while dynamic structure is defined at the coupled system level. Local structural changes of the continuous dynamics can be

modelled by structuring the dynamic description of the atomic DEVS. For that purpose logic variables inside the rate of change function f , the state event condition function c_{se} and the continuous output function λ_c can be used.

In order to allow dynamic structure behaviour on the coupled system level, e.g. the creation, deletion and exchange of submodels, extended specifications to represent this dynamics need to be established. Barros (Barros 1996) suggests to add a special atomic model called network executive to each coupled model which holds the structure information and describes the structure dynamics. In contrast to Barros’ approach Pawletta et al. (Pawletta, Lampe, Pawletta, and Drewelow 2002) propose to extend the classic DEVS coupled model as defined in (2) in a way that allows to hold structure information directly in the coupled model. This work takes the second approach for further analysis. The specification of a dynamic structure DEVS coupled model according to Pawletta et al. is as follows:

$$N_{dyn} = (X_N, Y_N, \{d_N\}, S_N, \delta_{Nx\&s}, \delta_{Nint}, \lambda_{Nd}, ta_N) \quad (6)$$

with the set of structure states S_N :

$$S_N = H_N \times \{M_d \mid d \in D\} \times EOC \times EIC \times IC \times Select \quad (7)$$

Table 3 lists extensions introduced for dynamic structure coupled systems. For meaning of identifiers which are same as for classic coupled DEVS models compare with table 2.

Table 3: Extensions for DSDEVS hybrid coupled systems

in coupled system N_{dyn}	
d_N	name of the coupled system
S_N	set of structure states
$\delta_{Nx\&s}$	external and state event transition function
δ_{Nint}	structure state transition function
λ_{Nd}	discrete output function
ta_N	time advance function
in set of structure states S_N	
H_N	set of structure related state variables

Extensions summarised above allow structure variability, which is prerequisite for the implementation of workpieces as atomic systems. Possible structural changes at the coupled system level are

- Creation,
- Cloning,
- Deletion and
- Replacement of atomic or coupled subsystems,
- Their movement between coupled systems and
- Changes of couplings between system components.

The actual composition of a subsystem set and its coupling relations is called structure state. A dynamic structure DEVS can have different structure states $s_0, s_1, \dots, s_n \in S_N$ as defined in (7). Coupling information as well as *Select* rules for dynamic structure coupled DEVS are capsuled in the set of structure states S_N . Furthermore, structure dynamics information, e.g. the number and kind of structure changes already achieved, needs to be stored. The set of structural variables H_N holds this information. For a dynamic structure coupled DEVS the *Select* function can depend on the structure state and structure dynamics information.

The functions $\delta_{Nx\&s}$, δ_{Nint} , λ_{Nd} , and ta_N provide operations analogical those for atomic systems and uplift dynamic behaviour to coupled system level. In analogy to event-oriented dynamic behaviour of classic atomic DEVS systems, dynamic structure changes in coupled DEVS are induced by events. Relevant events are external, internal or state events. If an external event occurs, it is sent to the affected subsystems, as known from classic DEVS. To check out if it influences the structure dynamics of the coupled system, its external and state event transition function $\delta_{Nx\&s}$ is executed. Afterwards the time advance function ta_N is called to recalculate the time until the next internal event of the coupled system. State events that affect structure can be caused by output events of subsystems or threshold events of (i) continuous outputs of subsystems, (ii) continuous inputs of the coupled system or (iii) structure related states of the set H_N . For state events also first $\delta_{Nx\&s}$ is executed, and then ta_N is called.

Changes in the subsystem set or their coupling relations are never carried out by the $\delta_{Nx\&s}$ function directly, since this could lead to ambiguous event handling because external events could influence subcomponents and the structure state simultaneously. To avoid the definition of a further rule to resolve such situations, it is appropriate to convert changes in the subsystem sets and coupling relations with only the structure state transition function δ_{Nint} . A coupled system's external and state event transition function $\delta_{Nx\&s}$ should only modify structure related variables in the set H_N to prepare for internal structure events.

Events influenced by the internal structure of a coupled system are planned by the time advance function ta_N , and the proposed structure changes are then accomplished with the structure state transition function δ_{Nint} . Simultaneous internal events are controlled by the *Select* function. For the generation of structure related output events caused by internal events, the discrete output function λ_{Nd} is introduced.

Presented specifications for atomic and coupled DEVS models together with a new simulation concept form the DSDEVS-hybrid formalism. The formal approach and its integration into a programmable scientific and technical computing environment (SCE) are described in detail by Pawletta et al. (Pawletta, Deatcu, Hagendorf, Pawletta and Colquhoun 2006).

4.2. Dynamic Structure Coupled Model of Oven

If dynamic structure models are used to model a heating facility, it is obvious to represent the oven by a coupled model and not by an atomic DEVS as it is shown in figure 2. As the oven is a section of a larger system and is able to hold up to three atomic models of type workpiece at a time, it provides an input through which atomic models enter the system. Relevant parts of the formal specification of the coupled model *OVEN1* are:

$$OVEN1 = (X_N, Y_N, \{d_N\}, S_N, \delta_{Nx\&s}, \delta_{Nint}, \lambda_{Nd}, ta_N) \quad (8)$$

$$X_N = \left\{ in_done, in_wp \mid \begin{array}{l} in_done \in \{unfinished, done\}, \\ in_wp \in workpieces \end{array} \right\}$$

$$Y_N = \{ out_wp \mid out_wp \in workpieces \}$$

$$d_N = \{ "OVEN1" \}$$

with the set of structure related state variables and the index set which are parts of the set S_N according to (7):

$$H_N = \{ state_number, number_wp, wp_passed, \dots, new_wp, wp_done, \sigma \dots \}$$

$$D = \{ "heating_rod" \}$$

with characteristic functions:

$$\delta_{Nx\&s}(X_N, e, \sigma)$$

%no state events
% ext. events of OVEN1 or of subsystems
manipulate H_N
 $\sigma = 0$ % activate structure event

$$= (H_N, \sigma)$$

$$\delta_{Nint}(H_N, \{M_d\}, EIC, EOC, IC, \sigma)$$

change structure based on H_N
manipulate H_N
 $\sigma = inf$ % deactivate structure event

$$= (H_N, \{M_d\}, EOC, EIC, IC, \sigma)$$

$$\lambda_{Nd}(out_wp) = (wp_done)$$

$$ta_N(\sigma) = (\sigma)$$

where σ is a variable to store the time until next structure event.

Some elements of the set of structure related states S_N are highlighted for selected system states in 4.3.

4.3. Flow of Atomic Models through the Oven

For getting insight into the mechanisms important for dynamic structure modelling, the coupled model of oven is further examined. Step by step the structure changes in the coupled model *OVEN1* starting with initialisation of the model are described.

At initialisation time, illustrated in figure 4, the coupled model *OVEN1* is empty and contains just one atomic system specifying the heating rod which controls the oven's temperature. Dynamic behaviour of the atomic system *heating_rod* is not examined further, since it is not relevant for comprehension of the concept of workpieces as integral parts of the system. It is

assumed that the heating rod tries to keep a temperature level and its temperature interacts with those of the workpieces.

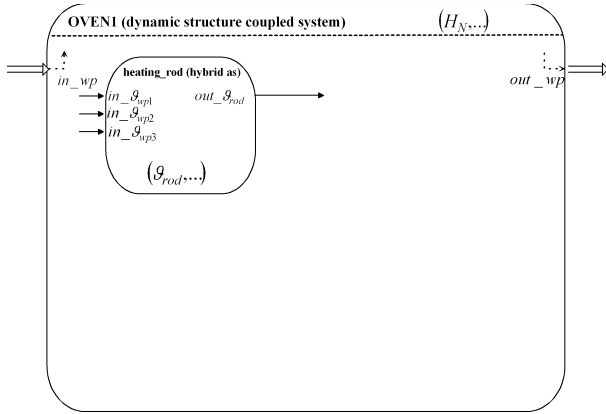


Figure 4: *OVEN1* at Initialisation Time

Dotted arrows, which point to the top, indicate that events occurring there induce structure changes. Dotted arrows, which come from the top of a coupled model, represent events that are results of structure changes. At initialisation time, the coupled model *OVEN1* provides the input *in_wp* and the output *out_wp* for workpieces. Events occurring there cause or are result of structure changes in the coupled model.

The structure of *OVEN1* at initialisation time is represented by the structure state $s_0 \in S_N$.

$$s_0 = (H_N, \{M_d \mid d \in D\}, EOC, EIC, EC, Select) \quad (9)$$

with:

$$H_N = \{state_number = 0, number_wp = 0, \dots \\ wp_passed = 0, new_wp = none, \dots \\ wp_done = none, \sigma = inf, \dots\}$$

$$D = \{ "heating_rod" \}$$

$$\{M_d \mid d \in D\} = \{M_{heating_rod}\}$$

$$EOC = \emptyset$$

$$EIC = \emptyset$$

$$IC = \emptyset$$

Select: empty function

The *Select* function which is called if simultaneous internal events occur in subsystems does not need to be specified, since there is just one subsystem at initialisation time.

With the arrival of the first workpiece an external event for the coupled model *OVEN1* is induced. The event *in_wp* occurs and the external and state event transition function $\delta_{Nx\&s}$ is executed, which manipulates the set of structure related state variables H_N . Relevant changes in H_N are (i) setting *state_number* to 1, (ii) setting *new_wp* to *wp2* and (iii) setting time until next internal event σ to 0. After that, the time advance function ta_N sets the next internal event time of coupled model *OVEN1* to σ , which equates current time, so that

an internal event occurs immediately. The internal event invokes the structure state transition function δ_{Nint} which adds the atomic system *wp1* to the coupled model *OVEN1*. The continuous temperature output of the heating rod *out_g_rod* is connected to the input *in_g_amb* of the workpiece. Furthermore, the discrete output *out_done* of the atomic model *wp1* is observed by the coupled model *OVEN1* to notice, if a workpiece demands to leave the oven due to reached temperature g_target . The resulting system structure is shown in figure 5.

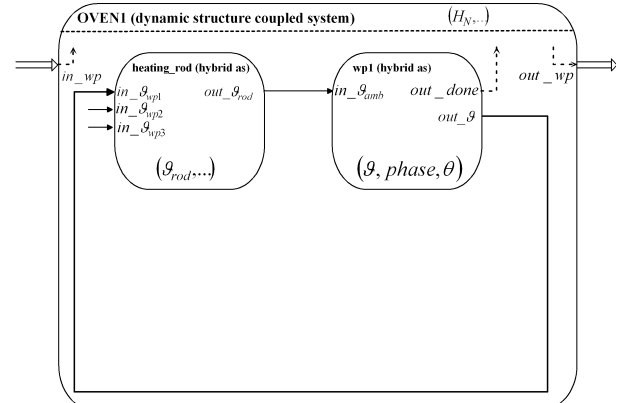


Figure 5: After Arrival of First Workpiece

The external and state event transition function $\delta_{Nx\&s}$ is always executed, if an external event at the input of coupled model *OVEN1* occurs or if an output event of any of the subsystems of *OVEN1* takes place. But just if the event influences the structure of the model, changes in H_N are carried out.

When all structure changes have been executed, the heating process starts. Continuous simulation cycles take place until they are interrupted by another workpiece arriving or the first workpiece asking to leave the oven due to reached target temperature.

In the figures 6 and 7 the structure of the coupled system *OVEN1* filled with two and three workpieces, respectively, is depicted.

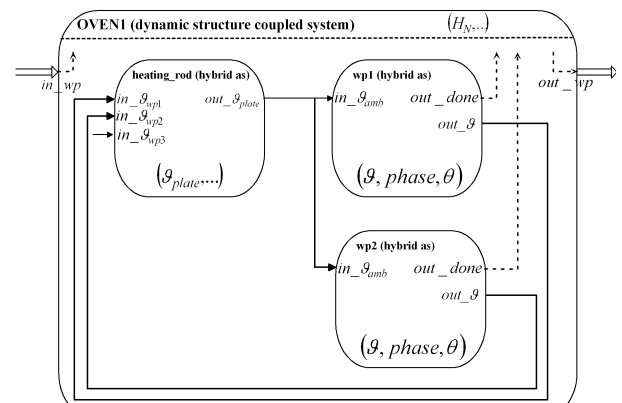


Figure 6: After Arrival of Second Workpiece

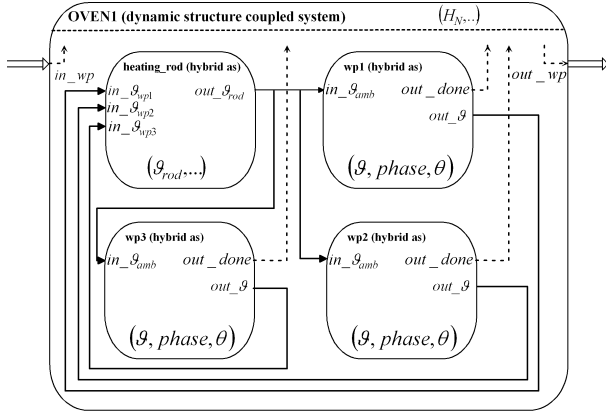


Figure 7: After Arrival of Third Workpiece

To point up sequential structure changes the associated structure states s_i , s_{i+1} and $s_{i+2} \in \mathcal{S}_N$ are analysed. Formal specification of the set of structure states \mathcal{S}_N is defined in (7). For the oven filled with two workpieces the structure state s_i can be described as follows:

$$s_i = (H_N, \{M_d \mid d \in D\}, EOC, EIC, EC, Select) \quad (10)$$

with:

$$H_N = \{state_number = i, number_wp = 2, \dots \\ wp_passed = 0, new_wp = none, \dots \\ wp_done = none, \sigma = inf, \dots\}$$

$$D = \{ "heating_rod", "wp1", "wp2" \}$$

$$\{M_d \mid d \in D\} = \{M_{heating_rod}, M_{wp1}, M_{wp2}\}$$

$$EOC = \emptyset$$

$$EIC = \emptyset$$

$$IC = ((heating_rod.out_g_{plate}, wp1.in_g_{amb}), \\ \dots (wp1.out_g, heating_rod.in_g_{wp1}), \\ \dots (heating_rod.out_g_{rod}, wp2.in_g_{amb}), \\ \dots (wp2.out_g, heating_rod.in_g_{wp2}))$$

Select : empty function

The *Select* function for simultaneous internal events, namely workpiece-done-events in case of simultaneously ready workpieces, can be left undefined, since order of workpieces leaving the oven is of no importance.

System state switches from s_i to s_{i+1} , when the third atomic model $wp3$ arrives, assuming none of the other two workpieces demands to leave the oven before. The arrival of $wp3$ is noticed as an external event of the coupled model *OVEN1*. Occurrence of this event invokes the external and state event transition function $\delta_{N_x \& s}$, which manipulates the variables in H_N .

It is important to remember that D , the set of submodels $\{M_d\}$, EOC , EIC , IC , and *Select* remain unchanged, because structure changes are just prepared in $\delta_{N_x \& s}$, but carried out later in the structure state transition function $\delta_{N_{int}}$.

$$s_{i+1} = (H_N, \{M_d \mid d \in D\}, EOC, EIC, EC, Select) \quad (11)$$

with:

$$H_N = \{state_number = i + 1, number_wp = 2, \dots \\ wp_passed = 0, new_wp = wp3, \dots \\ wp_done = none, \sigma = 0, \dots\}$$

$$D = \{ "heating_rod", "wp1", "wp2" \}$$

$$\{M_d \mid d \in D\} = \{M_{heating_rod}, M_{wp1}, M_{wp2}\}$$

$EOC, EIC, IC, Select$: unchanged

Based on the variable new_wp in H_N , σ is set to zero. Now the time advance function ta_N is called, which sets the time until next structure event to σ . Immediately, $\delta_{N_{int}}$ is executed and establishes the changes in the structure of the coupled model *OVEN1*. The resulting structure state s_{i+2} is as follows:

$$s_{i+2} = (H_N, \{M_d \mid d \in D\}, EOC, EIC, EC, Select) \quad (12)$$

with:

$$H_N = \{state_number = i + 2, number_wp = 3, \dots \\ wp_passed = 0, new_wp = none, \dots \\ wp_done = none, \sigma = inf, \dots\}$$

$$D = \{ "heating_rod", "wp1", "wp2", "wp3" \}$$

$$\{M_d \mid d \in D\} = \{M_{heating_rod}, M_{wp1}, M_{wp2}, M_{wp3}\}$$

$$EOC = \emptyset$$

$$EIC = \emptyset$$

$$IC = ((heating_rod.out_g_{rod}, wp1.in_g_{amb}), \\ \dots (wp1.out_g, heating_rod.in_g_{wp1}), \\ \dots (heating_rod.out_g_{rod}, wp2.in_g_{amb}), \\ \dots (wp2.out_g, heating_rod.in_g_{wp2}), \\ \dots (heating_rod.out_g_{rod}, wp3.in_g_{amb}), \\ \dots (wp3.out_g, heating_rod.in_g_{wp3}))$$

Select : empty function

Structure changes for the coupled model *OVEN1* can now just be caused by a workpiece asking to leave the oven. New workpieces arriving would be refused after evaluation of the structure related state variable $number_wp$.

5. SIMULATION OF DYNAMIC STRUCTURE HYBRID DEVS MODELS

Dynamic structure hybrid DEVS models are simulated by a modified discrete event simulation engine which calls an ordinary differential equation (ODE) solver between discrete event times. Structure information of the hybrid modular hierarchical model remains available during simulation time. Computation algorithms for modular hierarchical DEVS models including dynamic structure and hybrid system extensions were established in (Zeigler, Kim, and Praehofer 2000; Praehofer 1992; Barros 1996; Pawletta, Lampe, Pawletta, and Drewelow 2002). The key idea is to map a model specification to interacting program

objects for reflecting the system components and their coupling relations. For each coupled model a coordinator is employed, each atomic model has an associated simulator, so that for each part of the hierarchical model a program object exists which exclusively handles the dynamics, i.e. the simulation of this model part. These program objects are referred to as simulation objects of the computing model. On top of the hierarchically organised computing model the root coordinator initiates and controls the simulation cycles. Simulation is carried out by exchanging messages between the simulation objects.

Since for dynamic structure hybrid modelling hierarchical structure information needs to remain available during simulation, the model structure must not be flattened before simulation as it is usually done to speed up simulation. At the same time the employment of advanced ODE-solver methods requires the description of the continuous model equations and continuous state vectors of all model components in a closed form. Wrapper concepts can be used to provide the closed description.

The root coordinator known from classic DEVS simulation algorithms (Zeigler, Kim, and Praehofer 2000) is extended by an ODE-wrapper method including additional data structures to achieve a closed model description. These data structures hold by the root coordinator are (i) a vector of references to all continuous state variables and (ii) vectors filled with references to all atomic and coupled models. These references provide a dynamic representation of the modular hierarchical model in the required closed form. Taking benefit of the wrapper concept leads to possibilities for defining interfaces to advanced ODE solvers which are e.g. provided by programmable scientific and technical computing environments (SCEs).

For Matlab a prototype implementation of dynamic structure hybrid DEVS modelling and simulation was developed which is named *DSDEVS-hybrid toolbox*. Thus, practicability of the formal approach is proofed and verified. Major class definitions for the Matlab DSDEVS-hybrid toolbox reflecting the formal definitions for dynamic structure hybrid atomic and coupled DEVS models can be found in (Pawletta, Deatcu, Hagendorf, Pawletta and Colquhoun 2006).

6. CONCLUSION

Considering workpieces as integral parts of a model is of advantage if the workpieces or entities carry an own dynamic behavior. Modelling the dynamic of workpieces exactly where it appears prevents from having to adopt continuous descriptions distributed over the whole model. The process of modeling becomes more natural and realistic. Furthermore, reusability and maintenance of models is improved.

Workpieces which are moved through the system and are thereby temporarily integrated into system structures can be represented by atomic DEVS models adequately. If workpieces' dynamics is of hybrid nature,

hybrid atomic DEVS need to be employed. For the coupled DEVS model representing the overall system, e.g. a production line, dynamic structure behaviour is required.

REFERENCES

- Ashour, S. and Bindingnavle, S.G., 1973. An Optimal Design of a Soaking Pit Rolling Mill System. *Simulation* vol. 20: pp. 207-214.
- Barros, F.J., 1996. *Modeling and Simulation of Dynamic Structure Discrete Event Systems: A General Systems Theory Approach*. Thesis (PhD). University of Coimbra, Portugal.
- Cellier, F. E. and Kofman, E., 2006. *Continuous System Simulation*. New York/NY, USA: Springer.
- Hagendorf, O., Pawletta, T., and Deatcu, C., 2009. Extended Dynamic Structure DEVS. *Proceedings of EMSS 2009 - part of I3M Multiconference 2009*. September 23 – 25, Puerto de la Cruz, Spain
- Kofman, E., 2004. Discrete Event Simulation of Hybrid Systems. *SIAM Journal on Scientific Computing*. 25(5): pp 1771-1797
- Pawletta, T., Deatcu C., Hagendorf O., Pawletta S., and Colquhoun G., 2006. DEVS-Based Modeling and Simulation in Scientific and Technical Computing Environments. *Proceedings of DEVS Integrative M&S Symposium (DEVS'06) - Part of SpringSim'06*, pp 151-158. April 2-6, Huntsville/AL, USA.
- Pawletta, T., Lampe, B., Pawletta, S., and Drewelow, W., 2002. A DEVS Based Approach for Modeling and Simulation of Hybrid Variable Structure Systems. In: Engell, S., Frehse, G., and Schnieder, E., eds. *Modelling, Analysis, and Design of Hybrid Systems*. Berlin: Springer, pp. 107 – 130.
- Praehofer, H., 1992. *System Theoretic Foundations for Combined Discrete-Continuous System Simulation*. Thesis (PhD). VWGÖ, Vienna.
- Uhrmacher, A.M. and Arnold, R., 1994. Distributing and maintaining knowledge: Agents in variable structure environment. *Proceedings of 5th Annual Conference on AI, Simulation and Planning of High Autonomy Systems*, pp 178-194. San Diego/CA, USA.
- Zeigler, B.P., 1976. *Theory of Modeling and Simulation*. 1st ed. New York, USA: Wiley Interscience.
- Zeigler, B.P., Kim, T.G. and Praehofer, H., 2000. *Theory of Modeling and Simulation*. 2nd ed. San Diego/CA, USA: Academic Press.

AUTHORS BIOGRAPHY

Christina Deatcu, Thorsten Pawletta and Olaf Hagendorf are members of the research group Computational Engineering and Automation (CEA, <http://www.mb.hs-wismar.de/cea>) at Hochschule Wismar, located in North Germany close to the Baltic Sea.

Christina Deatcu is currently preparing her PhD thesis in cooperation with the University of Rostock. For the thesis the field of DEVS based modelling and simulation in scientific and technical computing environments is analysed. This includes the investigation of new formal approaches and algorithms for structure variable and hybrid DEVS systems. Furthermore she is active as a member of ASIM, the association for simulation in the German speaking area, and acts as co-organiser of workshops on modelling, simulation and control in automotive and process automation.

Dr.-Ing. Thorsten Pawletta is a professor of Applied Computer Science at the Faculty of Engineering at Hochschule Wismar. His research interests are focused on system theory of modelling & simulation, distributed modelling & simulation, modelling, simulation and control applications in engineering, and rapid control prototyping & robotics. Dr. Pawletta has published various papers in national and international journals as well as conference proceedings, and he is an active member of ASIM since 1991 and of SCS international since 1994.

Olaf Hagendorf received his diploma for electrical engineering from the University of Rostock in 1997. From 1997 to 2007 he managed a software company in Wismar and since then he works as a freelance engineer. His research area is structure optimisation of discrete event systems. He recently submitted his PhD thesis *An Approach for Simulation Based Structure Optimisation of Discrete Event Systems* which is result of a cooperation of the research group CEA, Wismar and the Liverpool John Moores University.

Dr.-Ing. habil. Bernhard P. Lampe is a professor of Automatic Control in the Institute of Automation at the University of Rostock, Germany. He is member of numerous scientific boards and committees and is known from various papers and several monographs. Main research areas are digital and sampled-data control systems with maritime, automotive and medical applications.

EXTENDED DYNAMIC STRUCTURE DEVS

Olaf Hagendorf ^(a), Thorsten Pawletta ^(b), Christina Deatcu ^(c)

^(a) Liverpool John Moores University, School of Engineering, UK

^(a, b, c) Hochschule Wismar, University of Applied Sciences: Technology, Business and Design, Germany

^(a) oh@ibhagendorf.de, ^(b) thorsten.pawletta@hs-wismar.de, ^(c) christina.deatcu@hs-wismar.de

ABSTRACT

Since the first publication of DEVS, the formalism was enhanced and many extensions have been introduced. Every extension holds some advantages over the other, e.g. Parallel DEVS generalizes the specification and handling of concurrent events, DEVS with Ports enables a more structured modeling and Dynamic Structure DEVS introduces dynamic structure changes at coupled model level during simulation time. The extensions have one joint attribute: they are extending the Classic DEVS formalism and don't incorporate the advantages of each other. Hence, the decision on one DEVS extension inhibits the use of advantages of another one. This lack leads to the idea of a merging formalism to combine the advantages of different approaches. The Extended Dynamic Structure DEVS combines the Classic DEVS with some of the existing extensions: Parallel DEVS, Dynamic Structure DEVS and DEVS with Ports.

Keywords: Discrete Event Simulation, DEVS, DSDEVS, PDEVS, EDSDEVS

1. INTRODUCTION

The DEVS formalism was first introduced by Zeigler (Zeigler 1976) in the 1970s. In (Zeigler et.al. 2000) the authors classify this formalism, position and compare it with other, more established modeling and simulation formalisms. Several international research groups are working on the DEVS formalism and are regularly publishing results at the annual DEVS Symposium at Spring Simulation Conferences, European Modeling and Simulation Symposia and others. Wainer (Wainer 2009) maintains a list of available DEVS tools. The DEVS formalism is, in contrast to other modeling and simulation formalisms, not very widely used in industrial practice. This situation persists despite the fact that the theory is a well-founded, general formalism. It can only be assumed that one reason of the marginal acceptance is the type of available software tools (Pawletta et.al. 2006).

There are several publications to extend the application field or to ease the use of DEVS e.g. Parallel DEVS generalizes the specification and handling of concurrent events, DEVS with Ports enables a more structured modeling and Dynamic Structure DEVS

introduces dynamic structure changes at coupled model level during simulation time and significantly eases the modeling of larger real systems. The extensions have one joint attribute: they are based on the Classic DEVS formalism and extending it in a specific direction. Hence, the decision on one DEVS extension inhibits the use of advantages and application fields of another one. This lack leads to the idea of a merging formalism to combine the advantages of different approaches and widen the application field of the resulting formalism.

The Classic DEVS formalism with the formal modeling concept and simulation algorithms is introduced in chapter 2. After a short introduction of a few DEVS extensions, three of them are described in detail in chapter 3. The fusion of Classic DEVS with the introduced extensions to the new Extended Dynamic Structure DEVS approach is presented with formal concept, simulation principles and algorithms in chapter 4. The conclusions in chapter 5 complete this contribution.

2. CLASSIC DEVS

DEVS is a modular, hierarchical modeling and simulation formalism. Every DEVS model can be described by using two different model types, atomic and coupled. Both model types have an identical, clearly defined interface through input and output ports. An atomic model describes the behavior of a non-decomposable entity via input/output events and event driven state transition functions. A coupled model describes the structure of a more complex model through the aggregation of several entities and their couplings. These entities can be atomic models as well as coupled models. The DEVS formalism consists of two parts: (i) a formal DEVS model definition and (ii) simulator algorithms.

2.1. Formal Concept

The formal Classic DEVS description defines coupled and atomic models as a combination of sets and functions. The description of an atomic model is a 7-tuple (Zeigler et.al. 2000):

$am = (X, Y, S, \delta_{ext}, \delta_{int}, \lambda, ta)$

- X, Y and S specify the sets of discrete inputs, outputs and internal states.
- $\delta_{ext}: Q \times X \rightarrow S$ where $Q = \{(s,e) \mid s \in S,$

$$0 < e < t_{next}$$

The external state transition function δ_{ext} handles external input events.

- $\delta_{int}: S \rightarrow S$
The internal state transition function δ_{int} establishes a new internal state.
- $\lambda: S \rightarrow Y$
The output function λ generates an output event depending on the internal state S .
- $ta: S \rightarrow \mathcal{R}_0^+ \cup \infty$
The time advance function ta schedules the time of the next internal event after each state transition.

Figure 1 shows the dynamic behavior of an atomic model.

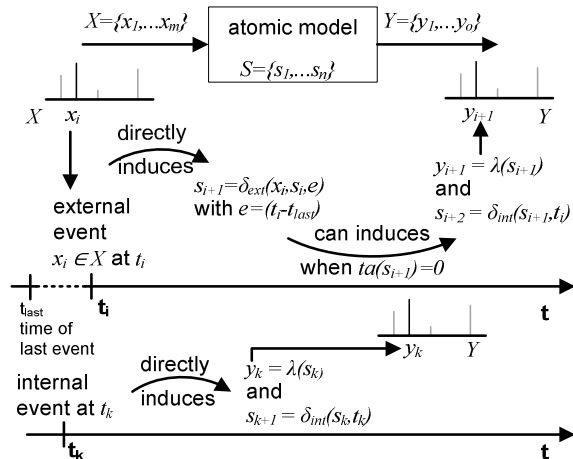


Fig. 1 Dynamic Behavior of an Atomic Model

The description of a coupled model is a 9-tuple (Zeigler et.al. 2000):

$$CM = (d_n, X, Y, D, \{M_d\}, EIC, EOC, IC, SELECT)$$

- d_n specifies the name of the coupled model.
- X and Y specify the sets of discrete inputs and outputs.
- D specifies the set of sub component names.
- $M_d \mid d \in D$
 M_d is the model of the sub component d
- EIC , EOC and IC are the sets of external input, external output and internal couplings.
- The $SELECT$ function prioritizes concurrent internal events of sub components.

Figure 2 depicts the relations of the elements of a Classic DEVS coupled model.

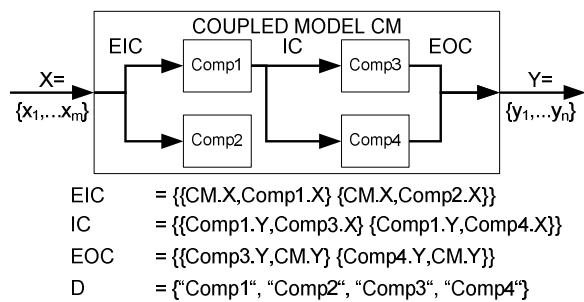


Fig. 2 Coupled Model Elements

The Classic DEVS approach supports the specification of behavioral system dynamics in atomic systems and the specification of static component aggregations in coupled systems. It is not possible to describe structural system dynamics at the coupled model level, i.e. the deletion or creation of components and couplings or changes of interfaces, although all necessary structural information is also available during simulation time. The only possibility to realize a structural system dynamic is to specify it with logical constructs at the atomic model level. However, this removes the advantages of reusability and model clarity and increases modeling complexity.

2.2. Classic DEVS Simulation

Beside the formal definition the second part of the Classic DEVS formalism is the description of abstract simulator algorithms for the execution of DEVS models. The algorithms are named abstract because they are implemented as a general pseudo code. The abstract simulator has a modular, hierarchical structure matching exactly the modular, hierarchical structure of a DEVS model. A DEVS model can be directly transformed into an executable simulator model using abstract simulator elements e.g. as shown in (Praehofer 1992; Zeigler et.al. 2000).

The abstract simulator approach consists of three different elements namely root coordinator, coordinator and simulator. Each atomic model is associated with a simulator element and each coupled model is associated with a coordinator element. The root coordinator is added to that structure as topmost ruling entity.

3. DEVS EXTENSIONS

Extensions of the Classic DEVS formalism increase the classes of system models that can be represented by DEVS. Several DEVS extensions are introduced e.g. in (Barros 1996; Chow et.al. 1994; Hagendorf et.al. 2006; Pawletta et.al. 1996; Praehofer 1992; Uhrmacher et.al. 1994; Wainer 2009; Zeigler et.al. 2000). An incomplete list of DEVS extensions recently presented is:

- DEVS with Ports: The port extension adds additional input and output ports to models.
- Parallel DEVS: Parallel DEVS (PDEVS) considers concurrent transition events.
- Dynamic Structure DEVS: Dynamic Structure DEVS (DSDEVS) enables changes during a simulation run. Several partial very different approaches exist. Dynamic structure extensions introduced by Barros (Barros 1996) and Pawletta (Pawletta et.al. 1996) keep the general structure of Classic DEVS modeling and simulation with additions to coupled model definitions but unchanged atomic model definitions. Other dynamic structure extensions e.g. an agent based DEVS (Uhrmacher et.al. 1994) introduce more extensive modifications.
- DSDEVS-hybrid: The extension of discrete state changes by continuous state changes as introduced by DSDEVS-hybrid enables a complete new

application field and can ease the modeling of several problems (Deatcu et.al. 2009).

- Real Time DEVS: The DEVS model is executed in real time rather than in model time. The time advance function delivers time intervals which allow uncertainty when an internal event has to take place.

The next sections introduce some of these DEVS extensions in more detail. They are used as basis of the subsequently introduced, unifying DEVS formalism.

3.1. DEVS with Ports

The introduction of ports into the Classic DEVS formalism makes modeling easier and the representation of information flow more clearly (Zeigler et.al. 2000). In Classic DEVS each model has only one single input and one single output port. All events are received and sent through these ports. With the port extension, a model has several input and output ports each dedicated for a specific task i.e. event type. A model can have several output ports which can be connected to input ports of other models as shown in figure 3. Hence, each event can use a dedicated, well defined routing path. The modeling becomes more structured; a model can become clearer and better understandable through differentiated interfaces.

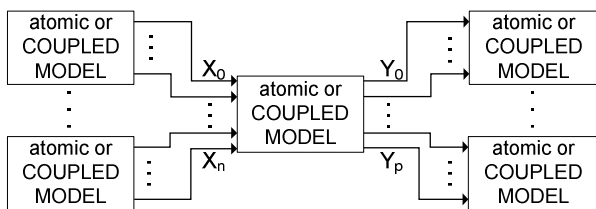


Fig. 3 Model with Multiple Input and Output Ports

The formal description of Classic DEVS with Ports largely remains the same except the extended definitions of X , Y for atomic and coupled models (Zeigler et.al. 2000):

$$X = \{(p,v) \mid p \in \text{InputPorts}, v \in X_p\}$$

$$Y = \{(p,v) \mid p \in \text{OutputPorts}, v \in Y_p\}$$

- p is the input or output port of the model
- v is a discrete value
- X_p / Y_p specify discrete inputs/outputs sets at port p

Whereas in Classic DEVS the coupling definitions consist of a sub model name as destination and source, respectively, for EIC and EOC and of a pair of sub model names for IC, the port extension necessitates a coupling definition extension, too:

- EIC = $\{(input_port, d.input_port) \mid input_port \in \text{InputPorts}, d \in D, d.input_port \in \text{InputPorts of } M_d\}$
- IC = $\{(d_i.output_port, d_k.input_port) \mid d_i, d_k \in D, d_i.output_port \in \text{OutputPorts of } M_{d_i}, d_k.input_port \in \text{InputPorts of } M_{d_k}, i <> k\}$
- EOC = $\{(d.output_port, output_port) \mid d.output_port \in \text{OutputPorts of } M_d, d \in D, output_port \in \text{OutputPorts}\}$

3.2. Parallel DEVS

Parallel DEVS (PDEVS) was introduced by Chow (Chow et.al 1994). It adds new elements and functions to the Classic DEVS formalism. It allows all imminent components to be activated simultaneously and enables sending their output to other components at the same time concurrently. Multiple outputs are combined in a bag which is sent as a whole to a model's external state transition function. A bag is similar to a set, containing an unordered set of elements, but allows multiple occurrences of an element. In Classic DEVS by contrast events are handled individually. In a PDEVS simulator (Zeigler et.al. 2000) during the *-message handling first all outputs are established before calling external and internal state transition functions. Each receiving component is responsible for examining and interpreting its combined inputs in the correct order. PDEVS gives the atomic model more control over the handling order of concurrent external and internal events. In Classic DEVS a super-ordinate component, the coupled model, is responsible for the execution order of concurrent internal events of different sub components using the *select* function. In PDEVS the order of simultaneous events is locally controllable at atomic model level with an additional, third state transition function, the confluent transition function δ_{con} . Hence, it merges the decision logic of execution order of concurrent events with the event handling functions at a same level.

According to the extensions of PDEVS an atomic model is defined by the following 8- tuple (Chow et.al. 1994):

$$am = (X, Y, S, \delta_{ext}, \delta_{int}, \delta_{con}, \lambda, ta)$$

- X, Y and S specify the sets of discrete input events, output events and sequential states.
- $\delta_{ext}: Q \times X^b \rightarrow S$ where X^b is a bag covering elements of X and $Q = \{(s,e) \mid s \in S, 0 < e < t_{next}\}$
The external state transition function δ_{ext} handles a bag covering external inputs $X^b = \{x_i \mid x_i \in X\}$.
- $\delta_{int}: S \rightarrow S$
The internal state transition function δ_{int} establishes a new internal state.
- $\delta_{con}: S \times X^b \rightarrow S$
The confluent transition function δ_{con} handles the execution sequence of δ_{int} and δ_{ext} functions during concurrent external and internal events.
 - The confluent function definition $\delta_{con}(s, X^b) = \delta_{ext}(\delta_{int}(s), 0, X^b)$ with $\delta_{ext}(s, e, X^b)$ is equivalent to the Classic DEVS behavior with a higher prioritized handling of internal events.
 - The alternative confluent function definition $\delta_{con}(s, X^b) = \delta_{int}(\delta_{ext}(s, ta(s), X^b))$ with $\delta_{int}(s)$ first handles external events.
 - The execution of the confluent function with an empty bag $\delta_{con}(s, null)$ calls directly the internal transition function δ_{int} .
- $\lambda: S \rightarrow Y^b$ where Y^b is a bag covering elements of Y

The output function λ generates a bag covering outputs $Y^b = \{ y_i \mid y_i \in Y \}$ depending on the internal state S .

- $ta: S \rightarrow \mathfrak{R}_0^+ \cup \infty$
The time advance function ta schedules the time of the next internal event after each state transition.

The definition of a coupled model for PDEVS is the same as for Classic DEVS except for the absence of the *select* function (Zeigler et.al. 2000):

$$CM = (d_n, X, Y, D, \{ M_d \}, EIC, EOC, IC)$$

The execution of a PDEVS model is carried out similarly to Classic DEVS with some changed details in the message handling (Zeigler et.al. 2000).

3.3. Dynamic Structure DEVS

Several approaches extend the Classic DEVS to Dynamic Structure DEVS (DSDEVS). Barros (Barros 1996) and Pawletta (Pawletta et.al. 1996) introduce two DSDEVS variants with an extension of the coupled model definition while the atomic model definition remains unchanged. With these extensions the coupled model is able to change its structure during simulation time. Uhrmacher (Uhrmacher et.al. 1994) introduces an agent based approach. It defines extensions for both atomic and coupled systems. Another approach is Cell-DEVS, a combination of cellular automata with the DEVS formalism where each cell consist of a single DEVS model (Wainer 2001).

The different types of extensions are carried out due to different application fields or problem definitions e.g. a typical Cell-DEVS application field is social and environmental modeling and simulation. The approaches of Barros and Pawletta are extending the classic formalism without changing its overall principle and thus without changing the general application field of Classic DEVS. The DSDEVS approach of Pawletta enables several options to specify structural dynamics:

- Creation, destruction, cloning and replacement of sub components
- Exchange of a sub component between two coupled models
- Changing coupling definitions of a coupled system

The DSDEVS approach extends the coupled model definition but the atomic model definition stays unchanged. During the simulation time a coupled model can change its structure. Each structure can be seen as a structure state s_i with $s_0, s_1, \dots, s_n \in S_{DS}$. A structure state s_i describes all structure relevant elements of a coupled model. Additionally a structural state set H_{DS} can store further structure information e.g. the number of structure changes at the present time or the current structure number. External or internal events, handled by the additional state transition functions $\delta_{x\&s}$ and δ_{int} at coupled model level, induce structure state changes and as a result model structure changes. This dynamic structure extension of Classic DEVS was developed with a regard to hybrid systems, i.e. systems with continuous and discrete event dynamic. In the following

only the relevant aspect for discrete event systems are taken into account.

A DSDEVS coupled model is defined by the following 6-tuple (Pawletta et.al. 1996):

$$CM_{DS} = (d_{ds}, S_{DS}, \delta_{x\&s}, \delta_{int}, \lambda, ta)$$

- d_{ds} specifies the name of the coupled model.
- According to the above definition of a coupled model, its structure consists of sets of sub components and coupling relations. Structure changes mean modifications of these sets. Obviously, the sets of sub systems and coupling relations could be interpreted as a structure state. The set of sequential structure states $\{s_0, s_1, \dots, s_n\} = S_{DS}$ defines all structure variants of the variable structure coupled model CM_{DS} . Structure state changes can be induced by handling external or internal events of the coupled model itself or by state events i.e. output events of subordinated components. A structure state is defined by a 9-tuple:

$$s_i = (X, Y, H_{DS}, D, \{ M_d \}, EIC, EOC, IC, select)$$

- X and Y specify the sets of discrete input and output events. The sets exactly match the sets X and Y in Classic DEVS. As an extension of DSDEVS the coupled model can directly handle external input events and can create external output events itself.
- The set H_{DS} represents additional structure related state variables. They are equivalent to the state variable set S of an atomic model.
- D specifies the set of sub component names.
- $M_d \mid d \in D$
 M_d is the model of the sub component d of the coupled model CM_{DS} . The set $\{ M_d \}$ defines all sub components of CM_{DS} .
- EIC , EOC and IC are the external input, external output and internal couplings.
- The function *select* prioritizes concurrent internal events of the coupled model itself and its sub components.

- $\delta_{x\&s}: Q_{DS} \times X \rightarrow H_{DS}$ where $Q_{DS} = \{(h,e) \mid h \in H_{DS}, 0 < e < t_{next}\}$

The external and state transition function $\delta_{x\&s}$ handles external input events and state events i.e. output events of sub components. However it is unreasonable to make changes in the set of sub components or the coupling relations by this function directly. This could lead to ambiguous event handling because external events could simultaneously influence the dynamic of sub components and the structure state. Consequently the $\delta_{x\&s}$ function is only allowed to modify structure related state variables in the set H_{DS} .

- $\delta_{int}: S_{DS} \rightarrow S_{DS}$
The internal transition function δ_{int} changes the structure state s_i to s_{i+1} and as a result induces a structure change of CM_{DS} . The execution of output function λ and internal transition function δ_{int} is

induced by a time driven internal event.

- $\lambda: S_{DS} \rightarrow Y$
The output function λ generates output events depending on the state S_{DS} .
- $ta: S_{DS} \rightarrow \mathfrak{R}_0^+ \cup \infty$
As with the dynamic of atomic models, internal events are scheduled by the time advance function ta . After each state transition the next internal event is established by the time advance function.

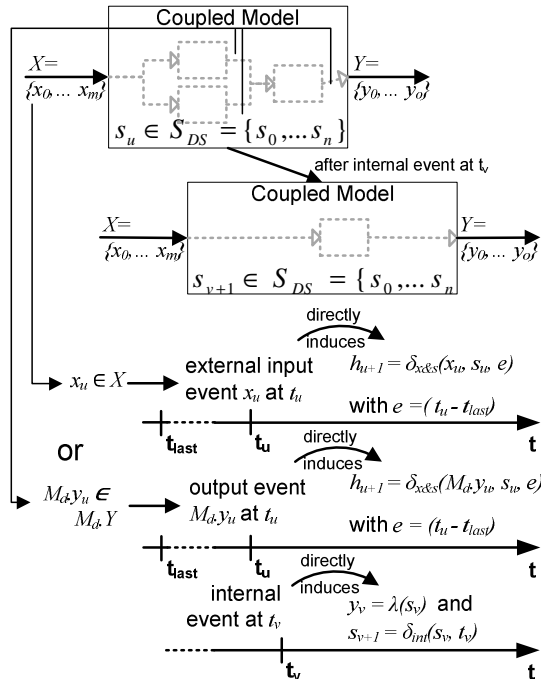


Fig. 4 Dynamic Behavior of a DSDEVS Coupled Model

The dynamic behavior of an atomic model is identical to the behavior in Classic DEVS. Figure 4 shows the dynamic behavior of a dynamic structure coupled model. The figure depicts two external input events and one internal event. Reasons for an input event handling can be an external input event at the input port of the coupled model itself or an external output event at the output port of a sub component M_d of the coupled model. The handling of both events by the coupled model is identically. As a result of an event the structure related state variable set H_{DS} can be changed and with the concluding call of the time advance function an immediate internal event can be induced. An internal event is handled by a coupled model similar to the internal event handling of an atomic model, i.e. the event handling can induce a change of the state sets S and S_{DS} , respectively.

4. EXTENDED DYNAMIC STRUCTURE DEVS

Chapters 3 and 4 introduce the Classic DEVS formalism and several DEVS extensions. This work aims to bring together all introduced approaches and to combine their advantages and application fields. In (Zeigler et.al. 2000) a first step into this direction is undertaken, the introduced PDEVS formalism is a combination of the original PDEVS and DEVS with

Ports. The Extended Dynamic Structure DEVS (EDSDEVS), proposed here, combines the extensions: Classic DEVS with PDEVS, DSDEVS and DEVS with Ports. The selection of the extensions is carried out to ensure the preservation of the generic modeling and simulation principles of Classic DEVS. The fusion results in a DEVS formalism with the following main characteristics:

- Modular, hierarchical and dynamic structure modeling and simulation formalism,
- Formal description by sets and functions,
- Exact definition of simulation algorithms,
- Dynamic behavior description in atomic models,
- Dynamic structure description in coupled models,
- Exact behavior definition of concurrent events,
- Substantial similarity between real system and model.

The next sections focus on the formal concept of EDSDEVS modeling with formal descriptions, dynamic behavior descriptions and introduction of the simulation concept with abstract simulator algorithms.

4.1. Formal Concept

The EDSDEVS formal descriptions of coupled and atomic models as a combination of sets and functions are structured similar to the Classic DEVS formal description. The EDSDEVS atomic model am_{EDS} is defined as an 8-tuple:

$$am_{EDS} = (X, Y, S, \delta_{ext}, \delta_{int}, \delta_{con}, \lambda, ta)$$

- $X = \{(p,v) \mid p \in \text{InputPorts}, v \in X_p\}$
 $Y = \{(p,v) \mid p \in \text{OutputPorts}, v \in Y_p\}$
The definitions of both sets are identical to the definitions in DEVS with Ports.
- S specifies the internal states set and is identical to set S of a Classic DEVS atomic model.
- $\delta_{ext}: Q \times X^b \rightarrow S$ with $X^b = \{x_i \mid x_i = (p,v), p \in \text{InputPorts}, v \in X_p\}$ and $Q = \{(s,e) \mid s \in S, 0 < e < t_{next}\}$
The external state transition function δ_{ext} handles a bag covering external inputs. Each input consists of a pair of a discrete input $v \in X_p$ and an input port $p \in \text{InputPorts}$. The set X_p is the set of discrete inputs at port p and InputPorts is the set of input ports of model am_{EDS} . The function δ_{ext} can induce an internal event with a rescheduling of the time of the next internal event. This extended definition of δ_{ext} is a fusion of the δ_{ext} definitions of PDEVS and DEVS with Port.
- $\delta_{int}: S \rightarrow S$
The internal state transition function δ_{int} can establish a new internal state. The execution of output function λ and internal state transition function δ_{int} is induced by a time driven internal event. The time of an internal event is established by the time advance function ta . The definition is identical to the definition in Classic DEVS.
- $\delta_{con}: S \times X^b \rightarrow S$
The confluent transition function δ_{con} handles the

execution order of δ_{int} and δ_{ext} functions during concurrent external and internal events. In spite of the same function signature $\delta_{con}(s, X^b)$ the parameter X^b is different to that in the PDEVS definition as described in section 3.2. Anyhow the three δ_{con} definitions from there also apply here. This extended definition of δ_{con} is based on the PDEVS δ_{con} function definition. Unlike in PDEVS the function has to handle a bag covering inputs, each consisting of a discrete input, input port pair.

- $\lambda: S \rightarrow Y^b$ whit $Y^b = \{y_i \mid y_i = (p, v), p \in OutputPorts, v \in Y_p\}$

The output function λ can generate a bag covering outputs Y^b . In spite of the same function signature $Y^b = \lambda(s)$ the function result Y^b is different to that in the PDEVS definition as described in section 3.2. The function result is a bag covering outputs $Y^b = \{y_i \mid y_i = (p, v)\}$ each consisting of a pair of discrete output $v \in Y_p$ and output port $p \in OutputPorts$. The set Y_p is the set of discrete outputs at port p and $OutputPorts$ is the set of output ports of model am . If and which outputs are generated depends on the internal state S . This extended definition of λ is based on the PDEVS λ function definition. Unlike in PDEVS the function generates a bag covering outputs each consisting of discrete output and output port pairs as introduced in DEVS with Ports.

- $ta: S \rightarrow \mathbb{R}_0^+ \cup \infty$
The time advance function ta schedules the time of the next internal event after each state transition. The definition is identical to Classic DEVS.

Figure 5 shows the dynamic behavior of an atomic EDSDEVS model am_{EDS} .

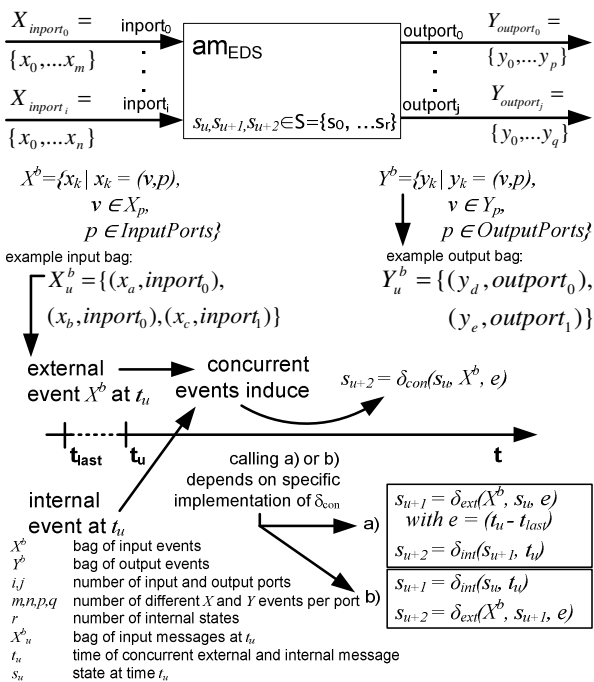


Fig. 5 Dynamic Behavior of an Atomic EDSDEVS Model

At time t_u the confluent transition function δ_{con} handles two concurrent events. The first event contains a bag covering external inputs received by the atomic model am_{EDS} . The figure depicts an example bag covering three external inputs received at two different input ports. A concurrent internal event at t_u was scheduled by the previous execution of the time advance function ta . Depending on the specific implementation of function δ_{con} sequence a) or b) is executed. The execution of λ creates a bag covering outputs. The shown bag Y_u^b covers two outputs.

An Extended Dynamic Structure DEVS coupled model is defined by the following 7-tuple:

$$CM_{EDS} = (d_n, S_{EDS}, \delta_{x\&s}, \delta_{int}, \delta_{con}, \lambda, ta)$$

- d_n specifies the name of the coupled model.
- In the EDSDEVS formalism the coupled model structure consists not only of sets of sub components and coupling relations as in DSDEVS but also of additional interface definitions i.e. input and output port definitions. The set of sequential structure states $\{s_0, s_1, \dots, s_n\} = S_{EDS}$ has to define all structure variants of the coupled model CM_{EDS} . Two model structure variants can vary in different interface definitions, in contrast to DSDEVS where each model has a non-variable interface with a single input and a single output port. Hence, a structure state has to incorporate interface definitions with sets of input and output ports additionally to the structure state definition as introduced in section 3.3. An EDSDEVS structure state is defined by a 10-tuple:

$$s_i = (X, Y, H_{EDS}, D, \{M_d\}, InputPorts, OutputPorts, EIC, EOC, IC)$$

- X and Y specify the sets of discrete input and outputs. The sets exactly match the extended definitions of X and Y as introduced in DEVS with Ports.
- The sets H_{EDS} , D and M_d exactly match the sets H_{VS} , D and M_d of the DSDEVS formalism introduced in section 3.3.
- $InputPorts$ and $OutputPorts$ specify the sets of input and output port names of the coupled model CM_{EDS} . These two elements of the structure state s_i are introduced by the EDSDEVS formalism.
- EIC , EOC and IC are the external input, external output and internal couplings of CM_{EDS} . The definition of the coupling relations exactly match the definition as introduced with the DEVS with Ports extension.

- $\delta_{x\&s}: Q \times X^b \rightarrow H_{EDS}$ where X^b is a bag covering input, input port pairs and $Q = \{(h, e) \mid h \in H_{EDS}, 0 < e < t_{next}\}$

The external and state transition function δ_{ext} handles a bag covering inputs, each consisting of a pair of a), b) or c):

- a) A discrete input $v \in Y_p$ and an input port $p \in InputPorts$. The set X_p is the set of discrete

inputs at port p and $InputPorts$ is the set of input ports of model CM_{EDS} .

- b) A discrete output $v \in M_d.Y_p$ and an output port $p \in M_d.OutputPorts$ where M_d is the model of the sub component d of the coupled model CM_{EDS} . The set $M_d.Y_p$ is the set of discrete outputs at port p and $M_d.OutputPorts$ is the set of output ports of model M_d .
- c) A discrete input $v \in M_d.X_p$ and an input port $p \in M_d.InputPorts$ where M_d is the model of the sub component d of the coupled model CM_{EDS} . The set $M_d.X_p$ is the set of discrete inputs at port p and $M_d.InputPorts$ is the set of input ports of model M_d .

This extended definition of δ_{ext} is a fusion and extension of the δ_{ext} definitions of DSDEVS, PDEVS and DEVS with Ports. In DSDEVS only state events induced by output events of sub components are handled. However, an output port can have coupling relations to multiple input ports. In this case there is a difference in the handling of a single output event of a single source sub model or multiple input events of different destination sub models. Hence, the external and state transition function of EDSDEVS can handle both output and input events. However, the functionality is in accordance with the description of the DSDEVS external and state transition function $\delta_{x\&s}$.

- $\delta_{int}: S_{EDS} \rightarrow S_{EDS}$
 $ta: S_N \rightarrow \mathfrak{R}_0^+ \cup \infty$
 The internal state transition function δ_{int} and the time advance function ta exactly match the functions of the DSDEVS formalism.
- $\delta_{con}: S_{EDS} \times X^b \rightarrow S_{EDS}$
 The confluent transition function δ_{con} handles the execution sequence of δ_{int} and δ_{ext} functions during concurrent external and internal events. The EDSDEVS formalism introduces the confluent transition function also at coupled model level due to the fusion of PDEVS and DSDEVS. An EDSDEVS coupled model handles external, state and internal events. Hence and in contrast to PDEVS, in EDSDEVS concurrent external and internal events can occur also at coupled model level. Consequently, a confluent transition function to handle concurrent events is necessary at this level. The functionality is in accordance with the description of the confluent transition function δ_{con} at atomic model level in this section.
- $\lambda: S_{EDS} \rightarrow Y^b$
 The output function λ generates a bag covering outputs $Y^b = \{y_i\}$ depending on state S_{EDS} . An output y_i consists of a pair of discrete output $v \in Y_p$ and output port $p \in OutputPorts$. The set Y_p is the set of discrete outputs at port p and $OutputPorts$ is the set of output ports of model CM_{EDS} . The output function λ in the EDSDEVS formalism merges three sources:

- The output function λ at coupled model level is introduced by DSDEVS.
- The definition of the function creating a bag covering outputs is based on PDEVS.
- The output event structure with pairs of output/output port is introduced by DEVS with Ports.

Figure 6 shows the dynamic behavior of a coupled EDSDEVS model CM_{EDS} . At time t_u the confluent transition function δ_{con} handles concurrent external and internal events. The first event is a bag covering inputs received at input ports by the coupled model CM_{EDS} . A concurrent internal event at t_u was scheduled by the last execution of the time advance function. Depending on the specific implementation of function δ_{con} sequence a) or sequence b) is executed. The execution of the internal state transition function δ_{int} can change the structure state s_u to s_{u+1} or s_{u+1} to s_{u+2} and therefore the model structure of CM_{EDS} to CM_{EDS}^* . The execution of the output function λ creates a bag covering outputs Y_u^b . The depicted example bag Y_u^b covers two outputs.

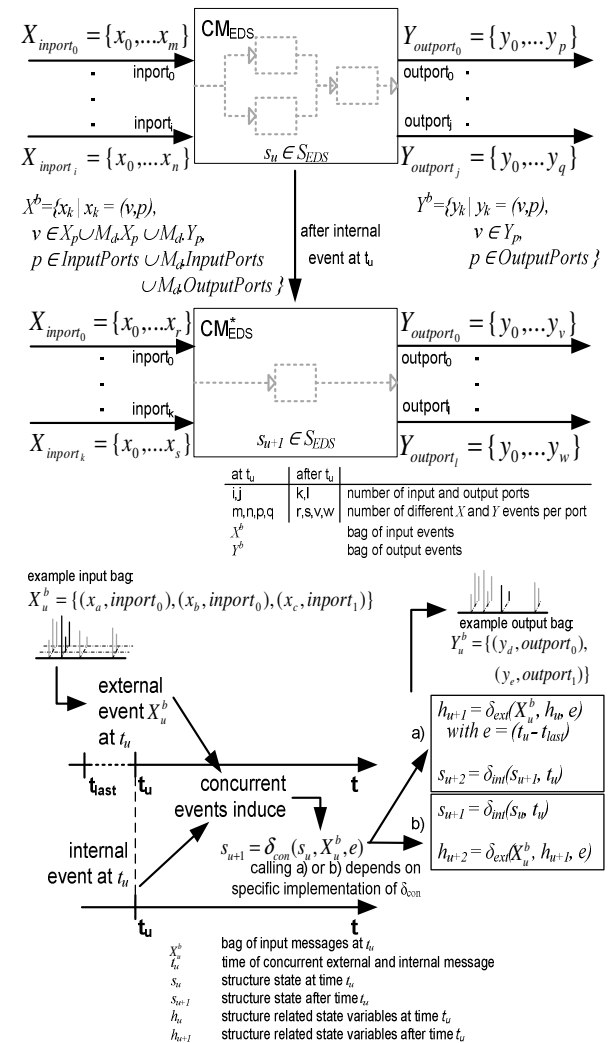


Fig. 6 Dynamic Behavior of a Coupled EDSDEVS Model

4.2. EDSDEVS Simulation

The simulation engine for EDSDEVS models is a combination and extension of the simulation algorithms of Classic DEVS, PDEVS and DSDEVS. The message handling of coordinators are largely similar to simulators. Each coordinator holds its own time of next internal event in t_{next_c} and searches the minimum time of next internal event in t_{next} of sub components and in its own t_{next_c} .

Figure 7 depicts an EDSDEVS model example with associated simulation model elements i.e. root coordinator, coordinator and simulator instances, message handling and model function calls. The overall structure is very similar to the Classic DEVS simulation model execution except for additions at the coordinators and associated coupled models. Because of complexity and clarity selected situations are shown in sections:

i. (Figure 7a) initialisation phase with i-message handling:

During the initialisation phase model component's init functions are called because of an i-message handling similar to Classic DEVS. Additionally, after structure changes during the simulation phase the init function is called too.

ii. (Figure 7b) *-message handling created due to an internal event of model $am2$:

The root coordinator advances the simulation clock and a *-message is firstly created. The message is sent to the successor coordinator instance of coupled model $CM1$ (not depicted). This coordinator instance compares the actual simulation time t with its own next internal event time stored in t_{next_c} and determines that it is not responsible for handling this event. Hence, the event is forwarded to the successor coordinator instance of $CM2$. The coordinator instance is again not responsible for handling the message itself but knows that a sub component scheduled the event. The coordinator instance will then forward the message to the appropriate simulator instance associated with $am2$. The simulator instance of $am2$ calls the model functions λ and δ_{int} . A result of calling λ could be a y-message sent back to the subordinate coordinator instance of $CM2$. This coordinator instance reacts with the call of the model function $\delta_{x\&s}$ of $CM2$ and a message forward to the simulator instance of $am3$ due to an appropriate IC coupling.

iii. (Figure 7c) *-message handling created due to an internal event of model $CM2$:

The depicted situation is similar to 7b except that the coordinator instance of $CM2$ determines that simulation time t and its t_{next_c} are equal. Hence, it has to handle the *-message itself with calling λ and δ_{int} model functions of $CM2$ with the possibility of generating a y-message sent to a sub component and/or superordinated coordinator instance and of changing its sequential structure state S_{EDS} .

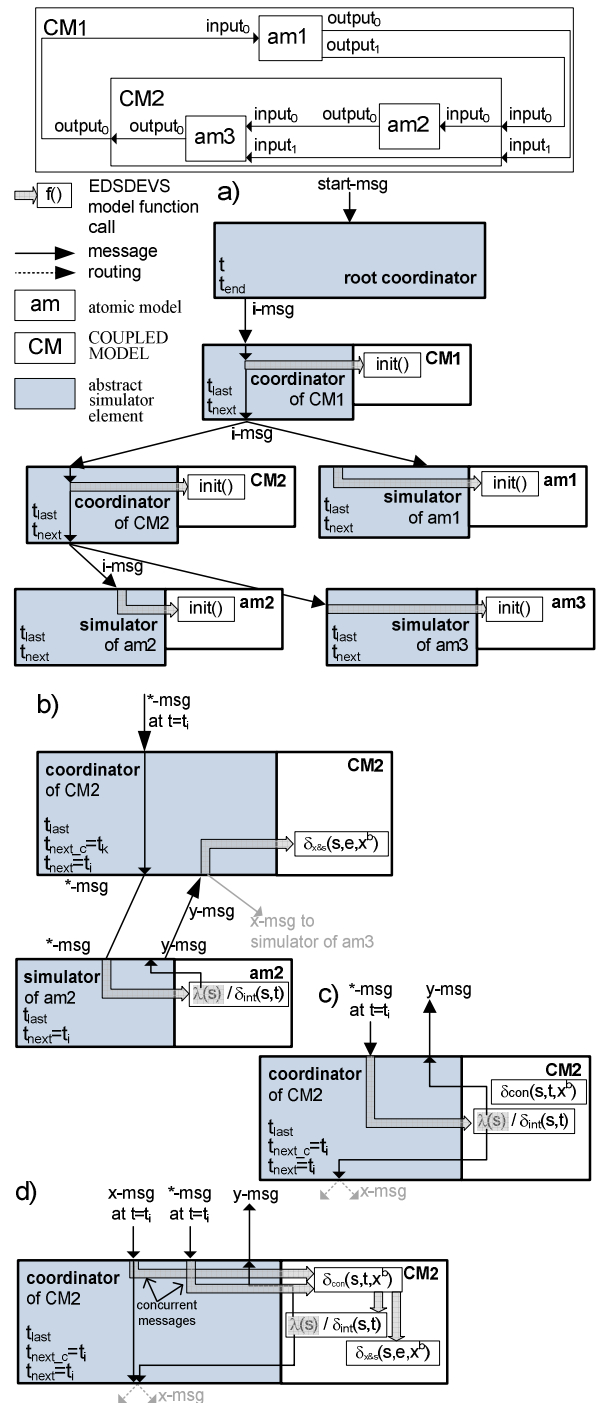


Fig. 7 EDSDEVS Model Example with Simulation Model Elements and Message Flow during Initialization and Simulation Phases

iv. (Figure 7d) concurrent event handling with the confluent transition function δ_{con} :

The figure depicts the handling of concurrent external and internal messages by the coordinator instance of $CM2$. The confluent function of $CM2$ is called to handle the concurrent messages. Depending on the specific implementation of δ_{con} the external transition function $\delta_{x\&s}$ and internal transition/output functions δ_{int} , respectively, are firstly called. The external message is concurrently

handled by the function δ_{con} and forwarded to the simulator instance of sub component *am2* as an x-message due to an appropriate EIC. Calling the output function λ could cause an y-message sent to a sub component and/or superordinated coordinator instance.

Listings 1 and 2 show the pseudo codes of the EDSDEVS simulator components.

```

variables:
  tlast // time of last event
  tnext // time of next int state event
  am // associated atomic model

// t=0 init at simulation start
// t>0 init after structure change
when receive i-msg(t) at time t
  am.init(t)
  tlast := t
  tnext := am.ta()

when receive *-msg(t) at time t
  if t <> tnext
    error: bad synchronisation
  ybag := am.λ()
  send ybag in a y-msg to parent coord.

when receive x-msg(t, xbag) at time t
  with value xbag containing x.value und
  x.port pairs
  if not (tlast ≤ t ≤ tnext)
    error: bad synchronisation
  if t=tnext and xbag is not empty
    //concurrent ext. & int. event
    am.δcon(t, xbag)
  else if t=tnext and xbag is empty
    // internal event
    am.δint(t)
  else
    // external event
    am.δext(t-tlast, xbag)
  end if
  tlast := t
  tnext := tlast + am.ta()

```

Listing 1 Pseudo Code of an EDSDEVS Simulator

```

variables:
  tlast // time of last event
  tnext // minimal time of next int.
  // state event of coupled model
  // or sub component
  tnext_c // time of next int state event
  //of the coupled model itself
  CM // associated coupled model with
  // CM.st current structure state
  IMM // imminent children
  mail // output mail bag

// t=0 init at simulation start
// t>0 init after structure change
when receive i-msg(t) at time t
  CM.init(t)
  foreach sub component Md ∈ CM.st.M
    send i-msg(t) to Md
  tlast := t

```

```

// determine time of next scheduled
// internal state event of coupled
// model itself
tnext_c := CM.ta()
// determine minimum time of next
// scheduled internal state events of
// coupled model and all subcomponents
tnext := min( tnext_c, { Md.tnext | Md
  ∈ CM.st.M } )

```

```

when receive *-msg(t) at time t
  if t <> tnext & t <> tnext_c
    error: bad synchronisation
  // internal state event of CM
  if t=tnext_c
    ybag := CM.λ()
    send bag of value/output port pairs
    in a y-msg to parent coordinator
  // internal state event of a subcomp.
  else if t=tnext
    // find all subcomps with tnext=t
    IMM:={Md | Md ∈ CM.st.M ∧ Md.tnext=t}
    foreach Md in IMM
      send *-msg(t) to Md

when receive x-msg(t, xbag) at time t
  with xbag containing x.value/x.port
  pairs
  if not (tlast ≤ t ≤ tnext_c)
    error: bad synchronisation
  if t=tnext_c and xbag is not empty
    // concurrent ext. and int. event
    CM.δcon(t, xbag)
  else if t=tnext_c and xbag is empty
    CM.δint(t) // int. event
  else
    CM.δext(t-tlast, xbag) //ext. event
  end if
  // get all subcomponents Md* with an
  // appropriate EIC
  receivers:=subcomponents{Md|Md∈CM.st.M}
  with {coupling|coupling∈CM.st.EIC}
  // forwards x-msg to all appropriate
  // subcomponents
  foreach subcomponent Md* in receivers
    // ext. event of subcomponent
    CM.δext(t-tlast, xbag)
    send x-msg(t, xbag, Md*.p) to Md*
    at port p
  foreach subcomponent Md* in IMM and not
  in receivers
    // send empty bag without inputport
    send x-msg(t, NULL, NULL) to Md*
  tlast := t
  tnext_c := tlast + CM.ta()
  tnext := min(tnext_c, {Md.tnext | Md∈CM.st.M})

when receive y-msg(t, ybag, d) at time t
  with ybag with value/port pairs from d
  // collect all y-msgs from all subcomp
  if d is not the last not reporting d
  in IMM
    add (ybag, d) to mail
    mark d in IMM as reporting
  // all subcomps now handled their *-msg

```

```

else if d is the last not reporting d
in IMM
  CM. $\delta_{\text{ext}}$ (t-tlast, mail)
  // check ext. coupling to form sub-
  // bag of parent output
  ybagparent = NULL
  foreach d in mail where (ybag and
  d) has an appropriate EIC
    add ybag to ybagparent
  send y-msg(t, ybagparent, CM) to
  parent model
  // check IC to get children Md*
  // with an appropriate IC who
  // receives a sub bag
  receivers := subcomponents{Md|d in
  mail, Md∈CM.st.M} with
  {coupling|coupling∈CM.st.IC}
  foreach subcomp Md* in receivers
    creates sub bag xbag from mail
    with elements where Md* is
    receiver
    send x-msg(t, xbag) to Md*
    mark d in IMM as sending
  foreach sub component Md* in IMM
    where Md* is not sending
    send x-msg(t, NULL) to Md*
tlast := t
tnext_c := tlast + CM.ta()
tnext := min( tnext_c, { Md.tnext | Md
∈ CM.st.M } )

```

Listing 2 Pseudo Code of an EDSDEVS Coordinator

5. CONCLUSIONS

The EDSDEVS formalism introduced in this contribution is a fusion of Classic DEVS with several extensions. This approach is as generic as possible modeling and simulation formalism based on DEVS. It widens significantly the application area of DEVS modeling and simulation. Further extensions are desirable and essential. To establish a widely accepted modeling and simulation approach extensions for parallel computing and graphical modeling are necessary. There are also approaches for hybrid DEVS extensions i.e. the support of continuous state changes. These proposals are recommended as further research.

REFERENCES

- Barros, F.J., 1996. *Modeling and Simulation of Dynamic Structure Discrete Event Systems: A General Systems Theory Approach*. Thesis (PhD), University of Coimbra
- Chi, S.D., 1997. Model-based Reasoning Methodology Using the Symbolic DEVS Simulation *Trans. of SCS*, 14(3): p.141-152
- Chow, A.C., Zeigler, B.P., 1994. Parallel Devs: A Parallel, Hierarchical, Modular Modeling Formalism. *Proceedings of the 1994 Winter Simulation Conference*, Lake Buena Vista/FL, USA
- Deatcu, C., Pawletta, T., Hagendorf, O., Lampe, B., 2009. Considering Workpieces as Integral Parts of a DEVS Model. *Proceeding of 2009 EMSS - part of I3M Multiconference 2009* Teneriffa, Spain

- Hagendorf, O., Pawletta, T., Pawletta, S., Colquhoun, G., 2006. An approach for modelling and simulation of variable structure manufacturing systems. *Proceeding of the 2006 ICMR* Liverpool, UK
- Pawletta, T., Lampe, B.P., Pawletta, S., Drewelow, W., 1996. Dynamic structure simulation based on discrete events. *ASIM-Mitteilungen Nr.53*, 9. *Workshop Simulation and AI*, p.7-11, Ulm, Germany, 02.1996.
- Pawletta, T., Deatcu, C., Pawletta, S., Hagendorf, O., Colquhoun, G., 2006. DEVS-Based Modeling and Simulation in Scientific and Technical Computing Environments *Proceedings of the 2006 Spring Simulation Conference*, Huntsville/AL, USA
- Praehofer, H., 1992. *CAST Methods in Modelling*. Pichler, F., Schwärtzel, H. Springer Pub.
- Uhrmacher, A.M., Arnold, R., 1994. Distributing and maintaining knowledge: Agents in variable structure environment. *5th Annual Conference on AI, Simulation and Planning of High Autonomy Systems*, p. 178-194
- Wainer, G., Giambiasi, N., 2001. Application of the Cell-DEVS paradigm for cell spaces modelling and simulation. *SIMULATION Transactions of The Society for Modeling and Simulation International*, vol. 76, 01.2001
- Wainer, G. A., 2009. *DEVS Tools*. Available from: www.sce.carleton.ca/faculty/wainer/standard/tools.htm [Accessed 06.2009]
- Zeigler, B.P., 1976 *Theory of Modeling and Simulation*. 1st edition, John Wiley
- Zeigler, B.P., Praehofer, H., Kim, T.G., 2000 *Theory of Modelling and Simulation*. 3rd edition, Academic Press

TSM: TEMPORAL SEQUENTIAL MACHINES

Norbert GIAMBIASI

LSIS: Laboratoire des Sciences de l'Information et des Systèmes,

University Paul Cezanne,

Marseille-France

norbert.giambiasi@lsis.org

Abstract

In this paper, we propose an extension of Sequential Machines by introducing a representation of time into these classical discrete event models. This new formalism, called TSM, is more expressive than Temporal Moore Machines proposed in (Giambiasi, N. 2009). It offers another step for a progressive design or analysis of discrete event systems: it is a new level with timed concepts in the hierarchy of formalisms starting from sequential machines going until DEVS and Timed Automata.

Keywords: sequential machines, DEVS, timed Automata.

1. INTRODUCTION

Several discrete event formalisms have been proposed for the modelling and the simulation of discrete event systems (Alur and Dill, 1994; Kohavi, 1980; Giambiasi, N., Escude, B. and Ghosh, S., 2000; Lynch and Vaandrager 1995; Zeigler, Praehofer and Kim, 2000; Zeigler, 1984). Now, an extensive theory is available on timed discrete event formalisms as the Discrete Event Specification: DEVS (Zeigler, 1984) and Timed Automata (Alur and Dill, 1994). These two classes of formalisms have an important expressive power allowing complex timed behavior to be specified, verified and simulated.

Less expressive timed formalisms can be useful in some classes of applications as design of manufacturing control systems, asynchronous circuit design, ... These less expressive formalisms can be used in a hierarchical design approach after a first untimed model specified by a sequential machine and before the specification of a complex timed behavior by a DEVS model or a Timed Automaton. In addition, this kind of formalisms offers a progressive approach for understanding the modelling concepts of discrete event systems.

In (Giambiasi, N. 2009), we have proposed an extension of Moore Machines by introducing a representation of time into these classical discrete event models. In this paper, we present a more expressive formalism by introducing timed behavior in the classical sequential machines. This formalism, called Temporal Sequential Machines (TSM) is more expressive than the previous one. In particular, TSM deals with discrete events

occurring in transitory states.

The paper is organized as follows:

In Section 2., we propose the definition of Temporal Sequential Machines and give the formal notions of executions or simulation runs of TSM models. In Section 3., we extend basic definition concerning sequential machines to TSM. In section 4., we demonstrate the closure under coupling of TSM models and give an example of coupled TSM.

2. TEMPORAL SEQUENTIAL MACHINES: TSM

TSM formalism is an extension of the Temporal Moore Machines (TMM) formalism (Giambiasi, N. 2009) in the sense that in a TSM model input events can occur in transitory states.

2.1 Definition

A Temporal Sequential Machine (TSM) is a structure:

$$M = \langle X, Y, S, \delta_{ext}, \delta_{int}, \lambda \rangle \text{ with:}$$

$X = \{x_i\}$, a finite set of input events,

$Y = \{y_i\}$, a finite set of output events,

$S = \{s_i\}$, a finite set of states with:

$$s_i = (p_i, \sigma_i) \text{ and } p_i \in V, \sigma_i \in \mathbb{R}_+^0 \cup \{\infty\},$$

p_i is the name of the state or of a class of states (states which differ by their lifetimes), σ_i is a state variable representing the lifetime of the state s_i . A state s_i with an infinite lifetime is said to be a steady (or stable) state, a state with a finite lifetime is a transitory state. We denote S_S the subset of steady states and S_T the subset of transitory states, then:

$$\forall \sigma_i = \infty, s_i \in S_S \wedge \forall \sigma_i \in \mathbb{R}_+^0, s_i \in S_T,$$

$$S = S_S \cup S_T \text{ and } S_S \cap S_T = \emptyset,$$

$\delta_{ext} : S_S \times X \rightarrow S_T$, the external transition function that specifies the state changes due to external events,

$\delta_{int} : S_T \rightarrow S_S$, the internal transition function that defines the state changes caused by an internal event corresponding to an autonomous behavior.

$\lambda : S_T \rightarrow Y$, λ is the output function only defined for transitory states.

As in DEVS formalism (Zeigler 1984), we introduce the concept of total state $q_i = (s_i, e)$ where e is the elapsed time in the current state s_i :

$$Q = \{(s_i, e), s_i \in S, e \in \mathbb{R}_0^+ \cup \{\infty\}\}.$$

The elapsed time in the state e is reset to zero for every discrete transition.

We introduce also the *lifetime* function that gives the lifetime of a state: $lifetime : S \rightarrow \mathfrak{R}_+^{0,\infty}$

2.2. Abstract Simulator and Operational behavior

The abstract simulator (figure 1.) runs with a coordinator, which sends to it the input events at the event time t_n . In this way this simulator gives the operational behavior or semantics of a TSM.

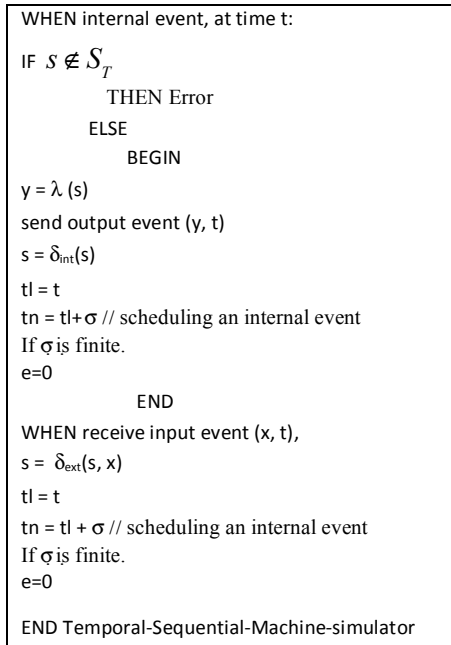


Figure 1: Abstract TSM simulator

2.3 State Transition Diagram and State Transition table of a TSM

A TSM can be represented as classical sequential machines by a state transition diagram or a transition table indicating the lifetime of the transitory states.

In the state transition diagram of a TSM the vertices represent the states with the value of the state lifetime and the edges are labelled with the input event or the output event associated with the transition.

In the state transition table of a TSM, for a stable state s_i and an input event x_i , the corresponding entry in the table defines $s_j = \delta_{ext}(s_i, x_i)$ and the lifetime of s_j , for a transitory state s_j , and the internal event the corresponding entry in the table defines $s_k = \delta_{int}(s_j)$, the lifetime of s_k and $\lambda(s_j)$.

2.4 Execution Fragments and traces of a TSM

We introduce the formal notion of executions of a TSM, and their traces as an alternative way of the abstract simulator to formally specify the full behavior of a TSM. These concepts are analogous to the ones that are commonly used for describing the behavior of Timed Automata (Lynch and Vaandrager 1995; Dacharry. H. and Giambiasi. N., 2005).

2.4.1. Definition: Execution fragment of a TMM model

An *execution fragment* of a TSM model $M = \langle X, Y, S, \delta_{ext}, \delta_{int}, \lambda \rangle$ is a finite alternating sequence

$Y = v_0 \, ev_1 \, v_1 \, ev_2 \, v_2 \dots ev_n \, v_n$, where:

1. pure time passage

Each v_i is a function from a real interval of the form $I_i = [0, t_i]$ to the set of total states of M, such that:

$$\forall j, j' \in I_i \mid j < j', \text{ if } v_i(j) = (s, e)$$

$$\text{then } v_i(j') = (s, e + j' - j)$$

2. discrete event transition

Each ev_i is an input or an output event, and if $(s_k, e) = v_{i-1}(sup(I_{i-1}))$, $(s_l, 0) = v_i(inf(I_i))$, one of the following conditions hold:

(a) $ev_i \in Y, \delta_{int}(s_k) = s_l$ with $lifetime(s_k) = e \wedge \lambda(s_k) = ev_i$

(b) $ev_i \in X, \delta_{ext}(s_k, x_i) = s_l \wedge e \leq lifetime(s_k)$.

2.4.2. Definition: Execution of a TSM.

Let $M = \langle X, Y, S, \delta_{ext}, \delta_{int}, \lambda \rangle$ be a TSM with an initial steady state s_i , $s_i \in S_s$ then an *execution* for M is a finite execution fragment of M that begins with s_i and finished with a state s_j .

We denote with $execs^*(M)$, $execs^\omega(M)$ and $exec(M)$

the sets of finite, infinite, and all executions of M, respectively.

We note:

- $state(\alpha)$ the set of discrete states that appear in the execution fragment α .

- $firststate(\alpha)$ and $laststate(\alpha)$ the functions that give the first state and the last state of a finite execution fragment α , respectively:

$$firststate(\alpha) = s_i \text{ if } v_0(inf(I_0)) = (s_i, 0)$$

and

$$laststate(\alpha) = s_i \text{ if } v_n(sup(I_n)) = (s_i, e).$$

2.4.3. Definition: Trace of a TSM.

Let $M = \langle X, Y, S, \delta_{ext}, \delta_{int}, \lambda \rangle$ be a TSM, α an execution fragment of M. Then, $trace(\alpha)$ is defined to be the tuple (θ_I, θ_O, w) such that θ_I and θ_O are sequences consisting of all the pairs of events, input and output respectively, of (α)

and their time of occurrences in (α) sorted in chronological order of occurrence, w is the total time of execution, defined as $\sum_{0 \leq j \leq n} \sup(I_j)$. Formally, the time of occurrence of an

event x_i of $\alpha = v_0 e v_1 v_1 e v_2 v_2 \dots e v_n v_n$ is equal to:

$$\sum_{0 \leq j < i} \sup(I_j), \text{ with } I_j \text{ the domain of } v_j$$

The set of all traces of a TSM is defined as

$$\text{traces}(M) = \{ \text{trace}(\alpha) \mid \alpha \in \text{execs}(M) \}.$$

3. TSM – BASIC DEFINITIONS

3.1. Definition: slow timed execution fragment

A timed execution fragment $\alpha = v_0 z_1 v_1 z_2 v_2 \dots z_n v_n$ of a TSM model is a *slow timed execution fragment* if:

$$\forall v_j \in \alpha \text{ with } v_j = (s_k, e) \wedge s_k \in S_T, \sup(I_j) = \text{lifetime}(s_k).$$

Remark: In a slow execution fragment, no external event occurs in a transitory state. In practice, every output event scheduled is realized. This behavior is analogous to the pure time delay behavior defined in the domain of digital circuit simulation (Gosh and Giambiasi, 2006).

3.2. Definition: Slow Input Trace

An input trace θ_i^j is a slow input trace iff:

$$\exists \alpha^i \mid \text{trace}(\alpha^i) = (\theta_i^j, \theta_O^i, t) \wedge \alpha^i \in \text{execs}^*_{\text{slow}}(M).$$

Notice that for a TSM model M which has only transitory states, $\forall s_i \in S, s_i \in S_T \wedge S_S = \emptyset$, the unique *slow input trace* is the *empty input trace* θ_i^\emptyset .

In this case, a slow execution fragment corresponds to an autonomous behavior.

3.3. Definition: Autonomous cycle of a TSM.

Let us consider a TSM model $M = \langle X, Y, S, \delta_{\text{ext}}, \delta_{\text{int}}, \lambda \rangle$ and an execution fragment α , if:

$$\forall s_i \in \text{state}(\alpha), s_i \in S_T \wedge \text{firststate}(\alpha) = \text{laststate}(\alpha),$$

α is an autonomous cycle execution fragment of the TSM.

3.4. Definition: Minimum cycle of a TSM.

Let us consider a TSM model $M = \langle X, Y, S, \delta_{\text{ext}}, \delta_{\text{int}}, \lambda \rangle$ with an autonomous cycle execution fragment α , α is an elementary cycle execution fragment of the TSM iff:

$$\forall v_i, v_j \in \alpha \wedge \forall i \neq j, v_i(\inf(I_i)) \neq v_j(\inf(I_j)).$$

A state can have only one occurrence in an elementary cycle execution fragment.

3.5. Definition: Completely specified TSM

A TSM model $M = \langle X, Y, S, \delta_{\text{ext}}, \delta_{\text{int}}, \lambda \rangle$ is *completely specified* iff: $\forall x_i \in X, \forall s_i \in S, \exists \delta(s_i, x_i) = s_j$.

A TSM is completely specified if, for every state, a transition is defined for every input event.

3.6. Definition: Deterministic TSM

A TSM model $M = \langle X, Y, S, \delta_{\text{ext}}, \delta_{\text{int}}, \lambda \rangle$ is deterministic and only if given an input trace θ_I (sequence of pairs of input events and their respective time of occurrence), its total time of execution t , and an initial state, s_k , there exists a *unique execution fragment* $\alpha = v_0 x_1 y_1 v_1 x_2 y_2 v_2 \dots x_n y_n v_n$ such that $\text{trace}(\alpha) = (\theta_I, \theta_O, t)$ and $v_0(0) = (s_k, 0)$.

3.7. Definition: Strongly connected TSM

A TSM model $M = \langle X, Y, S, \delta_{\text{ext}}, \delta_{\text{int}}, \lambda \rangle$ is *strongly connected* iff:

$$\forall s_i, s_j \in S, \exists \alpha \in \text{execs}^*(M) \text{ such as :}$$

$$\text{firststate}(\alpha) = s_i \wedge \text{laststate}(\alpha) = s_j$$

In other words, for every pair of states (s_i, s_j) of the TSM, there is an input sequence that taking the TSM from state s_i conduces it into the state s_j .

3.8. Definition: Distinguishable states.

Two states s_i and s_j of a TSM model M are distinguishable if and only if there exists at least two execution fragments α and β with: $\text{firststate}(\alpha) = s_i$ and $\text{firststate}(\beta) = s_j$ and $\text{trace}(\alpha) = (\theta_{I\alpha}, \theta_{O\alpha}, t_\alpha)$, $\text{trace}(\beta) = (\theta_{I\beta}, \theta_{O\beta}, t_\beta)$, such as:

$$\theta_{I\alpha} = \theta_{I\beta} \wedge \theta_{O\alpha} \neq \theta_{O\beta}.$$

The timed sequence $\theta_{I\alpha}$ (and $\theta_{I\beta}$) is called a distinguishing sequence of the pair (s_i, s_j) . If there exists for a pair (s_i, s_j) a distinguishing sequence of length k , then the states s_i and s_j are said to be k -distinguishable.

Let us remember that the input and output traces are sequences of timed events $(e v_i, t_i)$, and then two input or output traces θ_A and θ_B are equal if and only if:

$$\forall (e v_i, t_i) \in \theta_A \Leftrightarrow \exists (e v_j, t_j) \in \theta_B \mid t_i = t_j \wedge e v_i = e v_j.$$

3.9. Definition: k-equivalence of states.

Two states s_i and s_j of a TSM model M are k -equivalent if and only if they are not k -distinguishable.

3.10. Definition: State equivalence

Two states s_i and s_j of a TSM model M are equivalent if and only if they are k-equivalent $\forall k \in \mathbb{N}$.

3.11. Definition: Equivalence of TSM

Two completely specified TSM models $M1$ and $M2$ are *equivalent* if and only if, for every state in $M1$, there is a corresponding equivalent state in $M2$, and vice-versa.

3.12. Definition: Quasi-Equivalence of TSM

An incompletely specified TSM $M1$ is *quasi-equivalent* to a TSM $M2$, noted $M1 \leq_{qe} M2$, if and only if:

$$\forall s_i^1 \in S^{M1}, \exists s_j^2 \in S^{M2} \mid exec_{s_i^1}(M1) \subseteq exec_{s_j^2}(M2).$$

3.13. Definition: Minimal TSM

A TSM model is minimal (reduced) if and only if no two states in it are equivalent.

3.14. Example

Let us consider the barrel filling system of figure 2. It has four sensors that send-out the following events: *conveyor-stopped*, event that indicates the motor of the conveyor has been stopped, *barrel-full*, the barrel under the tank is full, *valve-closed*, the valve has been closed, *barrel-ok*, a barrel has been positioned under the tank.

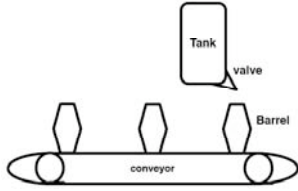


Figure 2: Barrel Filling System

This system under control has the following behavior:

When a barrel arrived under the tank, the conveyor is stopped and the system send out the event *conveyor-stopped* after 8 time units. Then, the valve is opened and 35 time units after the barrel is full, the event *barrel-full* is sent out. The control system closes the valve, and the event *valve-closed* is sent out after 4 time units. The process runs again by starting the conveyor. This behavior can be formally described with a TSM model (figure 3). The process can be stopped when the conveyor is running. It remains in an initial state "INIT" until a run-conveyor control is send to it, we suppose that during this time, the barrels are repositioned. Then, the TSM model of this system under control has the following set Y of output events:

$$Y = \{conveyor - stopped, barrel - full, barre - ok, valve - closed\}$$

the set X of input event is:

$$X = \{run - conveyor, stop - conveyor, open - valve, close - valve\}$$

and, the state set S is defined as follow: $S = S_s \cup S_T$, with

$$S_s = \{waiting1, waiting2, waiting3, waiting4, INIT\},$$

$$S_T = \{running, stopping1, stopping2, filling, closing - valve\}.$$

The transition functions, the output function and the lifetime (LT) of the states are given in the transition diagram of figure (only finite lifetime are indicated).

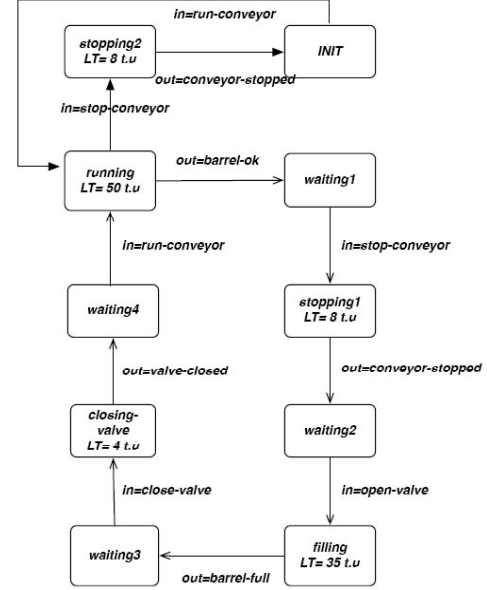


Figure 3. State Transition Diagram of the Filling System.

4. COUPLED TSM

4.1. Modular TSM models

In order to build complex TSM models we define the concept of coupled models in an analogous way than in DEVS formalism [14]. First, in order to have clean coupling relations, we define a modular TSM as a TSM with input-output ports through, which all the interactions with the environments must act. Ports are used to structure the input and output set of a TSM model M :

$$M = \langle X, Y, S, \delta_{ext}, \delta_{int}, \lambda \rangle, \text{ where,}$$

$$X = \{(n, v) / n \in N^X, v \in V\}, Y = \{(n, v) / n \in N^Y, v \in V\},$$

N^X and N^Y are the sets of input and output port names, v is the value of the event.

4.2. Modular Coupled TSM

Definition: A Modular Coupled TSM model is defined by a structure:

$$C = \langle X^C, Y^C, D^C, \{M^d\}, \{I^d\}, \{TR_{i,d}\}, Select^C \rangle$$

where:

X : finite set of input ports and values of the coupled model,

Y : finite set of output ports and values of the coupled model,

D : set of subcomponent names,

$\{M^d\}$: a set such that $\forall d \in D, M^d$ is a structure that defines a modular atomic TSM model,

$$M^d = \langle X^d, Y^d, S^d, \delta_{ext}^d, \delta_{int}^d, \lambda^d \rangle,$$

$\{I^d\}$: I^d is the *influencer* set of the subcomponent d :
 $I^d \subseteq D \cup \{C\} \wedge d \notin I^d$,

$\{TR_{i,d}\}$: $\forall d \in D \cup \{C\}, \forall i \in I_d, TR_{i,d}$ is an interface function define as follow:

i - $TR_{i,d} : X^C \rightarrow X^d$ if $i = C$,

ii - $TR_{i,d} : Y^i \rightarrow Y^C$ if $d = C$,

iii - $TR_{i,d} : Y^i \rightarrow X^d$ if $i \neq C \wedge d \neq C$

Select is the tie-breaking function to arbitrate the occurrence of simultaneous events in the coupled model.

In addition, we define the function $IMM(s)$ as the function that gives the subcomponents of C which are candidates for the next internal transition with:

$$s \in S, S = S^{d_1} \times \dots \times S^{d_i} \times \dots \times S^{d_n}, \forall d_i \in D :$$

$$IMM(s) = \{d \mid d \in D \wedge lifetime(s^d) = \min(lifetime(s^i)) \forall i \in D\}$$

4.3. Closure under coupling of TSM

An important property of the modular coupling of TSM is its closure. This guaranties that the coupling of TSM models defines a new equivalent TSM model. Therefore, a coupled TSM can be used as a component of another coupled TSM, allowing a hierarchical description of complex models.

Now, we show how a coupled TSM model defines an atomic TSM model.

The coupled model

$C = \langle X^C, Y^C, D^C, \{M^d\}, \{I^d\}, \{TR_{i,d}\}, Select \rangle$ with the components $M^d = \langle X^d, Y^d, S^d, \delta_{ext}^d, \delta_{int}^d, \lambda^d \rangle$ defines an atomic TSM model $M = \langle X, Y, S, \delta_{ext}, \delta_{int}, \lambda \rangle$ where:

$$X = X^C,$$

$$Y = Y^C,$$

$$S = S^{d_1} \times \dots \times S^{d_i} \times \dots \times S^{d_n}, \forall d_i \in D^C$$

The external transition function of the equivalent TSM is:

$$\delta_{ext}(s, x) = s', \text{ with } : s = (s_1, \dots, s_d, \dots, s_n) \wedge s' = (s'_1, \dots, s'_d, \dots, s'_n),$$

if $s'_d = \delta_{ext}(s_d, x_d)$ with $x_d = TR_{C,d}(x)$ then :

$$s'_d = \begin{cases} s_d^* & \text{if } C \in I_d \wedge x_d \neq \phi \\ s_d & \text{if not} \end{cases}$$

and, $lifetime(s) = \min(lifetime(s_d)) \forall d \in D$.

The internal transition function of the equivalent TSM is:

$$\delta_{int}((s_1, \dots, s_d, \dots, s_n)) = (s'_1, \dots, s'_d, \dots, s'_n) \text{ with}$$

$$s'_d = \begin{cases} \delta_{int}^d(s_d) & \text{if } d = Select(IMM(s)) \\ \delta_{ext}^d(s_d, x_d) & \text{if } d' \in I^d \text{ with } x_d = TR_{d',d}(\lambda^{d'}) \\ s_d & \text{if not} \end{cases}$$

The output function of the equivalent TSM is:

$$\lambda(s) = \begin{cases} TR_{d,C}(\lambda^d(s_d)) & \text{if } d \in I^C \wedge d = Select(IMM(s)) \\ \phi & \text{if not} \end{cases}$$

This demonstrates that the closure under coupling of TSM models is true.

4.4. Example

Let us consider a system with two cranes CR1 and CR2, and one wagon. CR1 loads the wagon at the beginning of the way and CR2 unloads the wagon at the end. There is a 'start/stop' input for the wagon. When a stop event is sent the wagon finish is run before stopping. The atomic models are given in figures 4 and 5.

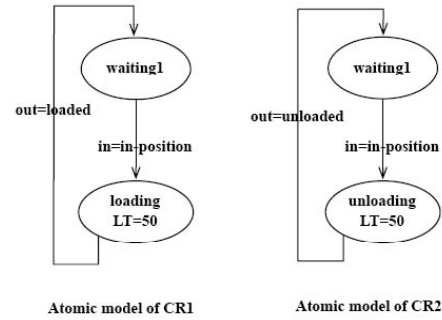


Figure 4. Atomic Models of the Cranes

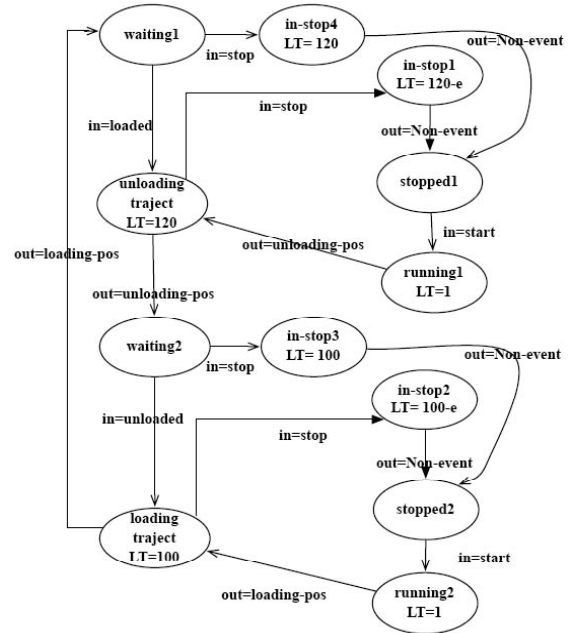


Figure 5: atomic TSM Model of the Wagon.

To build the model of the system we couple these three atomic models (figure 6). The coupled model has an input-output port to start or stop the whole system.

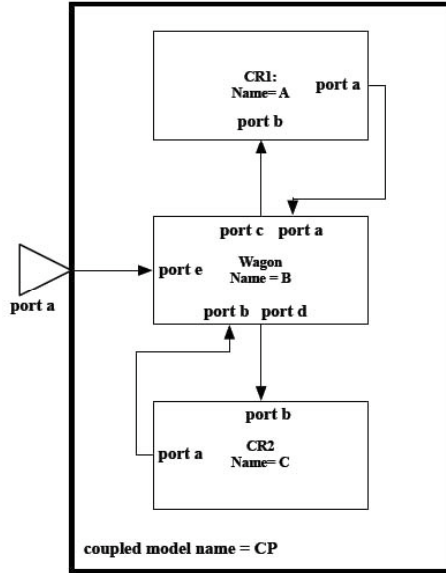


Figure 6. Coupled TSM model of the system.

The coupled model of the system

$CP = \langle X^{CP}, Y^{CP}, D^{CP}, \{M^d\}, \{I^d\}, \{TR_{i,d}\}, Select^{CP} \rangle$ is

formally defined as follow:

$$X^{CP} = \{start / stop\}, Y^{CP} = \emptyset,$$

$$D^{CP} = \{A, B, C\}, \{M^d\} = \{Cr1, Cr2, Wagon\},$$

$$I^A = \{B\}, I^C = \{B\}, I^B = \{A, C, CP\}$$

$$TR_{A,B}(port\ a, unloaded) = (port\ a, unloaded),$$

$$TR_{C,B}(port\ a, loaded) = (port\ b, loaded),$$

$$TR_{B,A}(port\ c, loading - pos) = (port\ b, in - position),$$

$$TR_{B,C}(port\ d, unloading - pos) = (port\ b, in - position),$$

$$TR_{CP,B}(port\ a, start) = (port\ e, start),$$

$$TR_{CP,B}(port\ a, stop) = (port\ e, stop).$$

5. CONCLUSION

In this paper, we have shown that the well-known sequential machines can be extended to represent timed behaviours and basic definitions from the field of sequential machines can be adapted to TSM

In addition, several results from Timed Automata theory can be also exploited in the problems of test and verification of TSM, and results from DEVS world on hierarchical modelling and simulation can also be used with TSM models.

REFERENCES

Alur. R. and Dill. D., 1994. A theory of timed automata. *Theoretical Computer Science*, 126:183–235, 1994.

Dacharry. H. and Giambiasi. N., 2005. From timed automata to devs models: Formal verification. In *Proceedings of SpringSim'05. SCS - The Society for Modeling and Simulation International*, 2005.

Lee, D. and Yannakakis, 1996. Principles and Methods of Testing Finite State Machines - A Survey. *Proceedings of the IEEE* 84. 1996.

Giambiasi, N., Escude, B. & Ghosh, S., 2000. Gdevs: a generalized discrete event specification for accurate modelling of dynamic systems, *Trans. of S.C.S.I.* 17-3: 120–134.

Kohavi, Z., 1980. Switching and Finite Automata Theory, *Computer Science Series. McGraw-Hill Higher Education*.

Lynch. N. A. and Vaandrager. F. W., 1995. Forward and backward simulations – part II: Timing-based systems. *Technical report, Laboratory for Computer Science, Massachusetts Institute of Technology, Cambridge, MA, USA, August 1995*.

Zeigler. B., Praehofer. H., and T. G. Kim., 2000. Theory of modeling and simulation, 2d edition. *Academic Press, London*.

Zeigler. Bernard P., 1984. Theory of Modelling and Simulation. *Krieger Publishing Co., Inc., Melbourne, FL, USA, 1984*.

Giambiasi. N. 2009. TMM: Temporal Moore Machines. *INCOM 2009, IFAC, Moscou, 5-7 juin 2009*.

S. Ghosh, N. Giambiasi., 2006. “A Novel Principle for Modeling and Simulation of Mixed-Signal Electronic Designs”, in: *IEEE Circuits & Devices*, November 2006.

AUTHOR BIOGRAPHY

Norbert GIAMBIASI is full Professor at the University of Aix-Marseille from 1981. In October 1987, he created a new engineer school and a research laboratory LERI in Nîmes (south of France). He was the Director of Research and Development in this engineer school. In 1994, he comes back to the University of Marseilles in which he creates a new research team in simulation. He creates a new CNRS laboratory (LSIS) in Marseilles with more than 180 researchers.

He has written a book on C.A.D and he is author of more than 250 international publications. He was and is scientific manager of more than 50 research contracts (with E.S Dassault, Thomson-Cimsa, Bull, Siemens, Cnet, Esprit, Euréka, Usinor, Phillips, ...). He was the research director of more than 45 PhD students. He is member of the program committee of several international conferences and he created international conferences ('Neural-network and their Applications', Conceptual Modelling and Simulation).

VALIDATING THE GLOBAL BEHAVIOR OF A SYSTEM DESCRIBED WITH SCENARIOS USING GDEVS AND Z.

Trojet wassim^(a), Sqali mamoun^(b), Lucile Torres^(c), Claudia Frydman^(d), Amine Hamri^(e)

^{(a) (b) (c) (d)}LSIS (laboratory of information sciences and systems)
University Paul Cézanne Aix-Marseille III
Avenue Escadrille Normandie-Niemen13397 MARSEILLE CEDEX 20

^(a)wassim.trojet@lisis.org, ^(b)mamoun.sqali@lisis.org, ^(c)lucile.torres@lisis.org, ^(d)claudia.frydman@lisis.org
^(e)amine.hamri@lisis.org

ABSTRACT

Interaction-based and state-based modeling are two complementary approaches of behavior modeling. The former focuses on global interactions between system components. The latter concentrates on the internal states of individual components. Both approaches have been proven useful in practice. One challenging and important research objective is to combine the modeling power of both for an automatic design synthesis. Initially, the system is described by a set of scenarios, each one is a partial view of the system. Automatic construction of specification from scenarios allows for the merging of partial behaviors into a global one. In fact, the synthesis process must produce a specification that includes all desired behaviors, with respect to static and temporal aspects. We present a combination of interaction-based and state-based modeling, namely, UML Sequence Diagrams and Discrete Event Specification, for the global system specification. We then propose the way to validate the overall system behavior derived from all scenarios using simulation techniques and formal methods in order to check consistency between requirements and the resulting specification. Our approach handles systems with intensive interactive behaviors as well as complex state structures.

Keywords: Scenarios, Synthesis, DEVS, Formal Methods.

1. INTRODUCTION

A typical development of an interactive system begins with writing scenarios which describe the most important behaviors. However there are no techniques (at the level of our knowledge) which allow directly the validation of the global system behavior described in the set of scenarios. Many approaches address the scenario synthesis problem and make possible to induce a total behavior models expressed in a State Machines (SM) format starting from a set of scenarios. For more information, the various approaches suggested are evaluated and discussed in (Aymot and Eberlein 2003) and (Liang et al. 2006). All of them agree with the need to have a sufficient number of scenarios. They

especially differ in the scenarios and target state machine notation, in their ability to support scenario composition mechanisms, in their use of merging techniques of identical states, and in the implemented synthesis algorithm. The idea of synthesizing State Machines (SM) out of a collection of scenarios has received a lot of attention in recent years. (Ziadi et al. 2004) and (Harel et al 2003) mentioned that a Sequence Diagram (SD) is an inter-objects view of a system, i.e. the history implying a cooperation of several objects to realize a functionality. In addition, designing a system behavior directly with SM is not an intuitive process, as the notion of state is often not natural in early stages of development, because SM are considered as an intra-objects description and are closer to an implementation.

In this paper, we use Sequence Diagram (SD) used in UML 2.0 (OMG 2007) to capture the behavioral requirements of a system. The SD is close to human understanding and usually remains abstract. Moreover, we propose to induce from a set of scenarios expressed in the form of SD, a discrete event specification (DEVS) (Zeigler et al. 2000) models representing the overall behavior of the system. We propose procedures for such transformation, and discuss the incoherencies problems between the behaviors of the scenarios models, and those of synthesized DEVS models. In fact, we aim to eliminate eventual ambiguities.

We have chosen DEVS because this formalism is powerful in modeling and simulation of discrete event systems, i.e., systems of which state change is due to the occurrence of events. So messages exchanged between objects of the SDs can be interpreted as events. Moreover, DEVS permits a hierarchical specification of the system. In fact, complex DEVS models are built from atomic models connected together in a hierarchical fashion. Interactions are mediated through input and output ports. Therefore, the behavior of each system object is modeled with an atomic DEVS model taking into account all scenarios containing it. Finally, the global behavior of the system is determined with coupling the atomic models in accordance with all interactions. The final coupled DEVS induced is used to validate the specification via simulation which should

produce the same sequence of events as the input sequences described in the scenarios. May some inconsistencies (conflicts, incoherencies, deadlock, ...) in the resulting DEVS models derived from scenarios could not be detected with simulation, in that case using formal methods (FMs) is necessary to detect and fix them. The integration of FMs in verification and validation process, through the use of formal verification techniques such as theorem proving, gives more credibility to scenarios correctness. The formal specification chosen for that end is the Z language. This choice is based on three reasons:

1. Z is a state/transition approach on which DEVS models are based.
2. Z is well known by the FMs community, so a lot of Z documentations are available.
3. Z is supported by many tools already available (Z/aves, Proofpower,...) enabling automatic formal verification.

In this paper, we assume that messages exchanged between objects hold several parameters. For that end we use the extension of DEVS called "GDEVS" (Giambiasi et al. 2000) which permits to specify systems depending on events with parameters. In fact, classic DEVS specify only atomic events, so input and output events do not have more than one value.

After brief preliminaries on scenario notations and on the DEVS formalism, we describe our approach by presenting the synthesis roadmap and by applying them to a server connection example. And finally we conclude the paper.

2. PRELIMINARIES

2.1 Scenarios notation

The scenarios are effective means to obtain and to validate the requirements. They became the most popular means to describe the system behavior. They describe how system components and users interact in order to provide system level functionality. Each scenario is a partial story which, when combined with all other scenarios, should conform to provide a complete system description. A great number of notations are commonly used for the description of scenarios, like: Message Sequence Charts (MSC) defined within an international standard (ITU 2000), Live Sequence Charts (LSC) proposed by (Damm and Harel 2001), the UML Sequence Diagrams (Erikson et al. 2004) and (OMG 2007), which are a simplified version of basic MSC... All of them are based on a textual and graphical representation. We use UML2.0 Sequence Diagrams (SD) to present our scenarios because (1) it is a formal language of which graphical notation is easily understood; and (2) it can be composed by using flow control operators (alternative, sequential composition, parallel composition, iteration, ...) in order to form more complex scenarios. The SD describes exchanges of messages (arrows) between communicating entities or

processes (called objects). Each object defines sequence of events, and is represented by a vertical axis. An event can be a message emission or reception.

A SD is a tuple $D = (E, \leq, O, \alpha, \phi)$ where:

- E : is a finite set of events divided into a set of sent events SE, and a set of received events RE;
- \leq : is a partial order relation (antisymmetric, reflexive and transitive) called causal order on events;
 - $\forall (e1) \in E \Rightarrow e1 \leq e1$ (reflexive);
 - $\forall (e1, e2) \in E^2, (e1 \leq e2) \wedge (e2 \leq e1) \Rightarrow e1 = e2$ (antisymmetric);
 - $\forall (e1, e2) \in E^3, (e1 \leq e2) \wedge (e2 \leq e3) \Rightarrow e1 \leq e3$ (transitive);
- O : is a set of objects names that perform at least one action in SC;
- α : is a set of action names;
- ϕ : $E \rightarrow O$ associates an event to an object. Moreover, events belonging to the same object are totally ordered;

$$\forall (e1, e2) \in E^2, \phi(e1) = \phi(e2) \Rightarrow (e1 \leq e2) \vee (e2 \leq e1)$$

The behavior represented by a D is a set of event sequences determined by the causal priority. This causal relationship determines a partial order, noted \leq , on the events between all objects. The partial order can be derived from D in respect with two principal rules:

- An event e drawn higher than another event e' on the same lifeline of an object precedes necessarily e' ;
- The event associated with a message sending precedes necessarily the event associated with the reception of this message (in the case of an asynchronous communication). For a synchronous communication, the events sending and reception for each message are used to be considered instantaneous.

We will denote by $em(e)$ the sending event corresponding to the receiving event e and $rec(e)$ the reception event corresponding to the sending event e . we use label $send(i, j, m)$ to denote the event " object i sends the message m to object j " and similarly, $receive(i, j, m)$ to denote the event " object i receives the message m from object j ". We will often note $!m$ the sending event, and $?m$ the receiving event for a message m .

2.2 Discrete Event System Specification

The DEVS formalism introduced by (Zeigler 1976) provides a means for modeling discrete event system in a hierarchical and modular way. The use of this formalism facilitates the modeling activity and guarantees a best accuracy of the model by decomposing the system into component models and by specifying the coupling

between them. There are two kinds of DEVS models, atomic models and coupled ones.

2.2.1 Atomic model

An atomic model (Zeigler et al. 2000) describes the behavior of a component, which is indivisible, in a timed state transition level. It is represented by one box comprising inputs and outputs; it allows a system to be described like a set of deterministic transitions between sequential states. Each transition is labeled by a sending or reception event from (Figure 1).

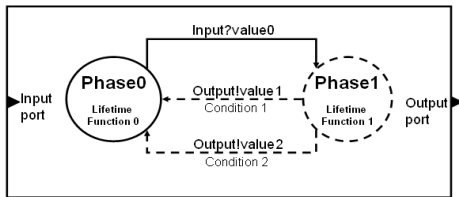


Figure 1: Representation of an atomic DEVS model

Formally, an atomic model is defined by a 7-tuple $\langle X, Y, S, \delta_{int}, \delta_{ext}, \lambda, ta \rangle$ where:

- X is the set of input values;
- Y is the set of output values;
- S is the set of sequential states;
- $\delta_{int} : S \rightarrow S$ is the internal transition function that defines the state changes caused by internal events;
- $\delta_{ext} : Q \times X \rightarrow S$ is the external transition function, where $Q = \{(s,e)|s \in S, 0 \leq e \leq ta(s)\}$ is the set of total state; this function specifies the state changes due to external events, with the ability to define a future state according to the elapsed time in the current state;
- $\lambda : S \rightarrow Y$ is the output function that generates output events;
- $ta : S \rightarrow \mathbb{R}_{0,\infty}^+$ gives the lifetime of the states, where $\mathbb{R}_{0,\infty}^+$ is the set of positive real numbers between 0 and ∞ .

2.2.2 Coupled model

The DEVS coupled model (Zeigler et al. 2000) is constructed by coupling the DEVS models. It can be composed by atomic models and/or coupled models through the coupling, output events of one model are converted into input events of the other. The coupled model (Figure 2) result can itself be employed as a component in a larger coupled model, by giving rise to the construction of complex models with hierarchical structures.

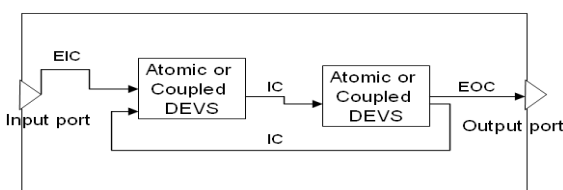


Figure 2: Representation of an atomic DEVS model

A coupled model is formally defined by a 7-tuple $\langle X, Y, M, EIC, EOC, IC, SELECT \rangle$ where:

- X is the set of input events;
- Y is the set of output events;
- M is the set of all the DEVS component models;
- $EIC \subseteq X \times \cup_i X_i$ is the external input coupling relation;
- $EOC \subseteq \cup_i Y_i \times Y$ is the external output coupling relation;
- $IC \subseteq \cup_i X_i \times \cup_i Y_i$ is the internal coupling relation;
- $SELECT : 2M - \phi, M$ is a function which chooses one model when more than 2 models are scheduled simultaneously.

2.2.3 GDEVS

Norbert Giambiasi generalizes the concept of discrete events modeling (Giambiasi et al. 2000). GDEVS defines event as a list of values. These values represent polynomial coefficients that approximate the input-output trajectory. Therefore, DEVS is a particular case of GDEVS (i.e. an order zero GDEVS). Formally, GDEVS represents a dynamic system as an n order discrete event model with:

$DEVSN = \langle XM, YM, S, \delta_{int}, \delta_{ext}, \lambda, D \rangle$. The following mappings are required:

- $XM = A^{n+1}$, where A is a subset of integers, string or real numbers
- $YM = A^{n+1}$
- $S = Q \times (A^{n+1})$

For all total state $(q, (a_n, a_{n-1} \dots a_0), 0)$ and a continuous polynomial input segment $w : \langle t1, t2 \rangle \rightarrow X$, are defined:

- The internal transition function :
 $\delta_{int}(q, (a_n, a_{n-1}, \dots, a_0)) = \text{Straj } q, x(t1+D(q, (a_n, a_{n-1} \dots a_0)), x)$, with $x = a_n t^n + a_{n-1} t^{n-1} + \dots + a_1 t + a_0$ and Straj model state trajectory $\forall q \in Q$ et $\forall w : \langle t1, t2 \rangle \rightarrow X, \text{Straj } q, w : \langle t1, t2 \rangle \rightarrow Q$
- The external transition function:
 $\delta_{ext}(q, (a_n, a_{n-1}, \dots, a_0), e, (a'_n, a'_{n-1}, \dots, a'_0)) = (\text{Straj } q, x(t1+e), x')$, with: $\text{Coef}(x) = (a_n, a_{n-1}, \dots, a_0)$ and $\text{Coef}(x') = (a'_n, a'_{n-1}, \dots, a'_0)$
- Coef: function to associates n -coefficient of all continuous polynomial function segments over a time interval $\langle ti, tj \rangle$, to the constants $(n+1)$ values $(a_n, a_{n-1} \dots a_0)$ such as:
 $w(t) = a_n t^n + a_{n-1} t^{n-1} + \dots + a_1 t + a_0$
 $\text{InCoef} : \lambda(q, (a_n, a_{n-1} \dots a_0)) = \Lambda(q, x)$
- The output function: $\lambda : S \rightarrow A^{n+1}$
- The function defining the life time of the states:

$D(q, (a_n, a_{n-1}, \dots, a_0)) = \text{MIN}(e/\text{Coef}(\text{Otraj } q, x(t1)) \neq \text{Coef}(\text{Otraj } q, x(t1 + e)), \text{ with Otraj model output trajectory: Otraj } q, w: \langle t1, t2 \rangle \rightarrow Y$

2.3 The Z specification language

Z (Woodcock and Davies 1996) is a state-based formal specification language based on the established mathematics of set theory and first-order logic. The set theory used includes standard set operators, set comprehension, Cartesian products, and power sets. Z has been used to specify data and functional models of a wide range of systems, including transaction processing systems and communication protocols. It has been standardized by ISO (ISO 2002). In Z, mathematical objects and their properties are collected together in schemas: patterns of declaration and constraint. The schema language is used to structure and compose descriptions: collating pieces of information, encapsulating them, and naming them for reuse. A schema contains a declaration part and a predicate part. The declaration part declares variables and the predicate part expresses requirements about the values of the variables.

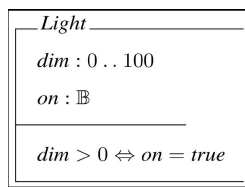


Figure 3: the Z schema of the light

The declaration part contains the declaration of two variables. The variable *dim* represents the illumination of the light object, which is of value from 0 to 100 (in percent) and *on* is a Boolean variable indicating whether the light object is on or not. The predicate part, referred to as the state invariant, places a constraint on the values of the two variables, i.e., the *dim* is nonzero if and only if the light object is on. A Z specification typically consists of a number of state and operation schemas. A state schema groups together state variables, e.g., the Light schema. An operation schema defines the relationship between the “before” and “after” valuations of one or more state schemas. Z is capable of specifying open system, i.e., systems which constantly interact with their environment. External inputs to an operation schema are denoted as variables followed by a question mark. Example: The operation schema defines the operation Adjust by how the state variables of the Light schema are updated.

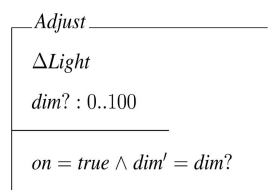


Figure 4: The Z operation schema to adjust the light

The schema name after Δ indicates the state schemas to be updated by the operation. The variable *dim?* is an input from the environment. The state-update is expressed using a predicate involving both primed and unprimed state variables. In particular, the operation can only be applied when the light is on and, after the operation; the light level is set to be dim. If *dim?* is zero, the state variable *on* will be set to false because of the state invariant.

3 STATE OF THE ART

Working in parallel with the requirements of a system expressed in the form of scenarios, and its specification provided by the state machines improves the level and quality of specification. The idea of synthesizing state-based models from interaction-based specifications has received much attention in recent years. Many approaches address the scenario synthesis problem and make possible to induce a total behavior model expressed in a state machine format starting from a set of scenarios (Liang et al. 2006). The synthesis process not only helps to significantly reduce the effort of system construction, but it also provides a bridge over the gap between requirements and implementation.

Figure 5 shows the different types of interaction-based to state-based model transformation.

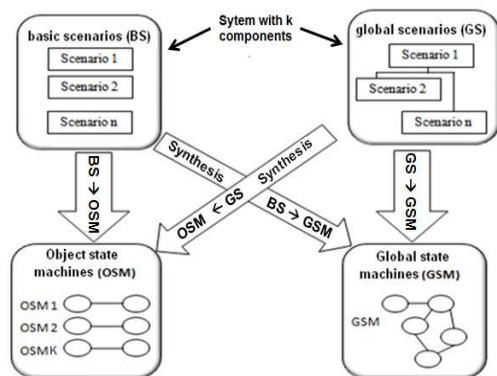


Figure 5: Different types of synthesis process

In the upper half of the diagram, interaction-based models are sub-categorized into either basic scenarios “BS” without using any composition mechanisms, or global scenarios “GS” obtained through the composition of BSs. In the lower half, we have two categories of state-based models: object state machines “OSM” and global state machine “GSM”. The arrows in the diagram show four direct transformation paths from scenarios to State Machines models via synthesis (i.e., BS→OSM, BS→GSM, GS→OSM, and GS→GSM). Each object state machine OSM represents the behavior of one object of the system. For the synthesis “BS→ OSM” and “GS→OSM”, the global behavior of the system described by the scenarios is defined with parallel composition of all the obtained state machines and which coordinate via shared messages.

In general, the existing synthesis approaches differ depending on:

- the choice of the scenarios language;
- their semantic interpretation;
- the type of target state machines;
- the complexity of the implemented synthesis algorithm;
- and the using of reunification techniques to regroup the identical states.

(Helouet and Jard 2000) have presented an approach for deriving automatically a communicating finite state machines (CFSMs for short) from hierarchical message sequence chart (HMSC), each process is described by a finite state machine that can send or receive messages. The traces of synthesized system are compared with traces allowed by the HMSC. If the set of traces are equivalent, then the protocol is considered correct. In this approach, an erroneous protocol can be constructed from a specification containing a non-local choice (Arjan et al. 2005) and potentially leading to deadlock. (Harel et al. 2005) proposed a synthesis approach using the scenario-based language called Live Sequence Charts (LSC) as requirements, and synthesizing a collection of finite state machines. (Letier et al. 2005) presents a technique to generate Labeled Transition System (LTS) from high Level Message Sequence Chart (hMSC), in this approach, complex system behavior is modeled by parallel composition of the component LTS models. (Ziadi et al. 2004) proposes an idea to synthesize state charts starting from combined scenarios expressed by basic UML2.0 Sequence Diagrams, and gives an algorithm for regrouping the state chart of the same object. Furthermore, (Damas et al. 2006) has presented an approach to generate Labeled Transition System from a collection of basics MSC's, and use a technique to merge the identical states. However, the existing synthesis techniques only focus on generating (global or local) state machines from scenario-based specifications, while the coordination among the behavior components in the systems is not considered. In our approach we use a novel synthesis technique, which can be used to generate, in the first hand, the behavior of each object of the system and, in the second hand, the relation between all the behaviors obtained to observe the global system behavior using GDEVS coupling. By using GDEVS, we can also recover the parameters values given in the messages exchanged between objects..

To that end, we have chosen the indirect synthesis of GSM from GS via OSM ($GS \rightarrow OSM \rightarrow GSM$). We use combined SD to represent GS, atomic GDEVS models to represent OSM, and Coupled GDEVS model to represent GSM. The advantage of using this transformation path (combined SD \rightarrow atomic GDEVS \rightarrow coupled GDEVS), is to make possible the simulation of objects behaviors (described in the GDEVS atomic model) and the global system behavior represented by the GDEVS coupled model. In addition GDEVS modularity gives flexibility to the synthesis process. Indeed, the latter is not influenced by any modification, addition or removal of a

system object. The final GDEVS model can be obtained by coupling the various atomic models.

4 TRANSLATING SEQUENCE DIAGRAMS INTO GDEVS

In this section we discuss a general procedure for deriving GDEVS components descriptions from a set of UML2.0 sequences diagrams (with or without composition mechanism). We extend the definition of the synthesizing algorithm of Ziadi's approach (Ziadi et al. 2004) to include the TimeEvent, conditions, and parameters of messages exchanged between the objects. While the SD captures interactions between a set of objects, the atomic GDEVS represents the internal behavior of a single object, which is essential to develop an executable simulation model. We follow the Ziadi's approach to synthesize the coupled GDEVS model by combining basic SDs into one combined SD. The coupled GDEVS is obtained from merging the various atomic GDEVS models obtained for each object of the system.

4.1 Roadmap for the translation procedure

We assume given a set of SDs that describe all the interaction sequences among a set of objects. Then, we aim to obtain an atomic GDEVS model for exactly one object "O", occurring in the SDs. Therefore, the global behavior of each object "O" is modeled with a global atomic GDEVS model taking into account all scenarios containing it. Finally, the global behavior is determined with coupling the atomic models obtained in accordance with all interactions. The procedure for obtaining those automaton consists of seven successive phases:

- **Verification:** This phase consists in checking that the whole of the behaviors described by each SD is a sequence set of events respecting the causal priority. The events associated with one object are totally ordered.
- **Projection:** During the second phase, we project each of the given SDs onto the object O, i.e. we remove all other instance axes, as well as message arrows that neither start, nor end at O.
- **Normalization:** Then, we normalize the projected SDs. we identify the events which will make possible to determine the initial and finale phases of the atomic GDEVS models corresponding to an object O.
- **Transformation into an atomic GDEVS:** this phase consists in translating reception events of the object into external transitions in GDEVS models and sent events into internal transitions. In the definition of the external transition function, $p?v$ notes the value v of the input event occurring on the input port p of the atomic model. In the definition of the output function, $p!v$ notes the value v of the output event to be generated on the output port p . If there are conditions, we add conditions in the equivalents states. If the message is sending event and contains parameters (Figure 6), the corresponding is internal transition $p!v$ with p is the name of receiver object, and v is EVENTVAL (name of the message sent,

followed by the set of parameters labels {parameter1;...;parameterN}. Else, if the message is received event, the corresponding is external transition $p?v$ with p is the name of sender object, and v is the set of parameters values received by the sender object {EVENTVAL[1];...;EVENTVAL[N]}. For this, each parameter is considered as state variable stored in the following state as it is shown in the following example:

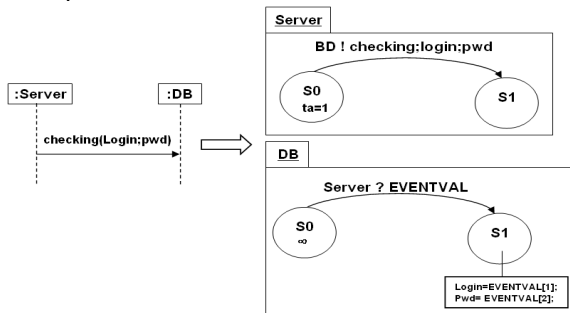


Figure 6: Message with parameters in atomic GDEVS

By implementing our algorithm synthesis using Ziadi's approach, "if an event in a SD has a Time Event TE , the corresponding state of the atomic GDEVS has the same Time Advance ta . If TE is not explicitly specified and in the case of a receiving event, the corresponding state of the atomic GDEVS waits forever ($ta: = infinite$) until the message is arrived. And if TE is not explicitly specified and in the case of a sending event, we assume that the corresponding state of the atomic GDEVS has ($ta: =1$) in order to define explicitly the causal order between events belonging to the same objects axis". This rule is defined in the following rule:

Rule 1: $TE \text{ Translate}(SD, Atomic_GDEVS)$

- *Input:* SD (sequence diagram with TimeEvent TE)
 - *Output:* AM_GDEVS (atomic GDEVS model with TimeAdvance ta)
 - if (TE in SD) then ta in AM_GDEVS := a time value of TE ;
 - else if (receiving event) then ta in AM_GDEVS := infinite;
 - else if (sending event) then ta in AM_GDEVS :=1;
- EndRule1.**

Normally, in this phase, each object must have the same number of atomic GDEVS as the number of SD's.

- Merging all atomic models obtained for each object into one global atomic model: for each object, we merge all resulting atomic models associated to each SD, into one global atomic model that represents the total behavior of the object in the system. To that end, we merge states that receive the same events from the same states:

- case of an internal transition: if $\delta_{int}(S_i) = S_j$ with $\lambda(S_i) = (p!v)$ and $\delta_{int}(S_i) = S'_j$ with $\lambda(S_i) = (p!v)$ then $S_j=S'_j$;

- case of an external transition: if $\delta_{ext}(S_i, e, p?v) = S_j$ and $\delta_{ext}(S_i, e, p?v) = S'_j$ then $S_j=S'_j$.

- Optimization: To optimize the resulting global atomic GDEVS models, we used standard algorithms of optimization from automata theory to make our models deterministic and to reduce to the minimum the number of states and transitions. In this phase, we merge states that receive and send the same events, also states which have the same state variables.
- Generating the global coupled GDEVS: The final GDEVS coupled model can be obtained by coupling the various global atomic models for each object. In that end, if an object O1 sends an event to another object O2, we connect the output port of O1 with the input port of O2, and vice versa. The final coupled GDEVS resulting describes the overall behavior of the system.

By using GDEVS, we can recover the parameters values given in the messages exchanged between objects. To validate the specification of the system behavior from the final coupled GDEVS model, we use the LSIS-DME tool developed within our laboratory (Hamri and Zacharewicz 2007). From a model and a set of data, the simulator provides the simulation results. For the moment, our approach is limited to three main operators of UML2.0 sequence diagrams (seq, alt and loop). The extension of this framework to include more UML 2.0 operators such as opt (the optional operator) is currently under study. An algorithm and construction principles of coupled DEVS model starting from several combined SD (with the notion of sequence, iteration and alternative) were presented in (Sqli and Torres 2008) and (Sqli et al 2008). The next section provides an example application of the procedure we have outlined here.

5 CASE STUDY AND SIMULATION

In this section we apply the translation procedure as described in Section 4.1 to the SD depicted in Figure 7.

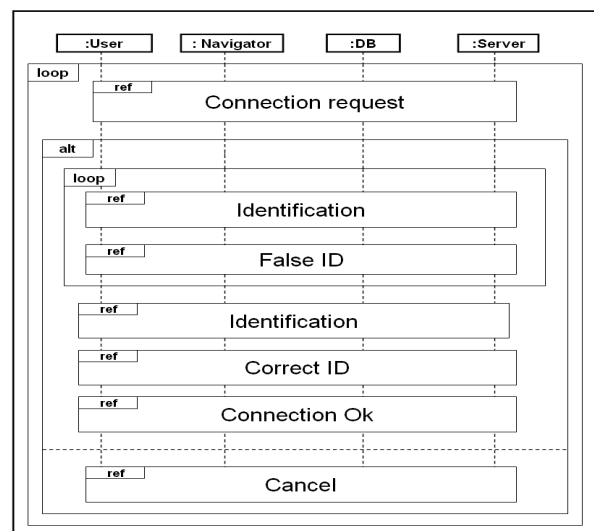


Figure 7: A SD representing a connection to the server with multiple scenarios

This example shows how a user can connect himself to a server, by starting a navigator. The user must enter the URL. After that, the navigator asks the user to be identified. If the password and the login of the user are corrects (Login="L_admin" and Pwd="P_admin"), connection will be established, else, the user receives an error message. Figure 8, 9, 10, 11, 12 and 13 represent the various scenarios which compose the connection server sequence diagram.

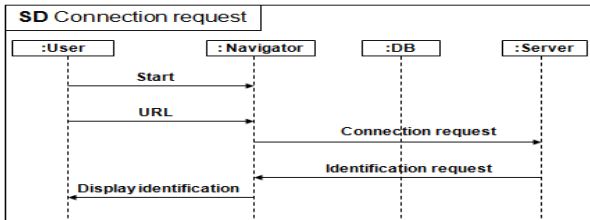


Figure 8: Connection request scenario

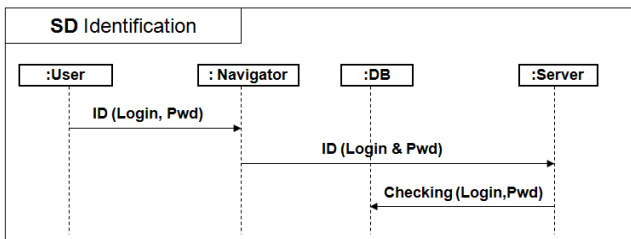


Figure 9: Identification scenario

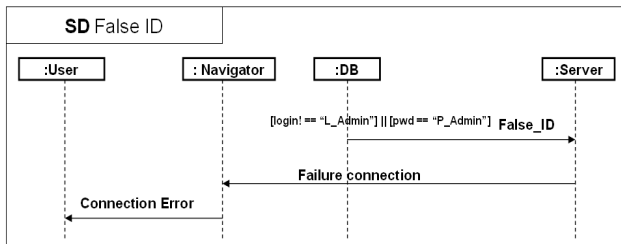


Figure 10: False ID scenario

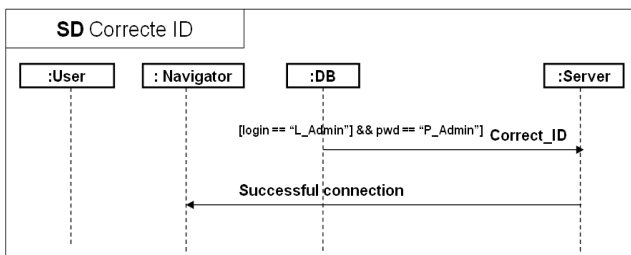


Figure 11: Correct ID scenario

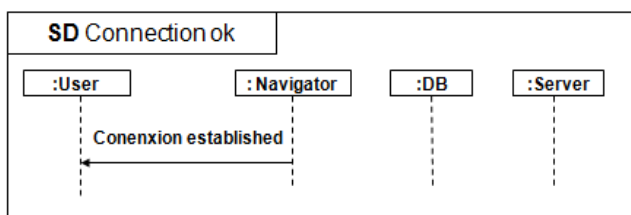


Figure 12: Conexion Ok scenario

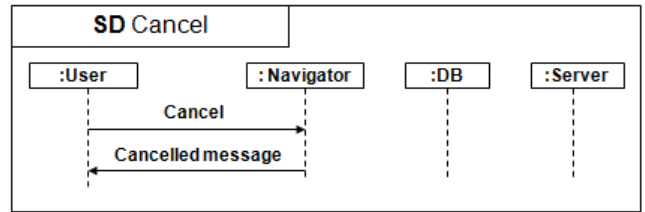


Figure 13: Cancel scenario

Figures 14, 15 and 16 shows the automatically synthesized global atomic model which specifies the behavior of each object of the system. By using states variables and GDEVs, we can recover the values of the login and the password sent with the parameters, and to compare them with the data stored in the database related to the server.

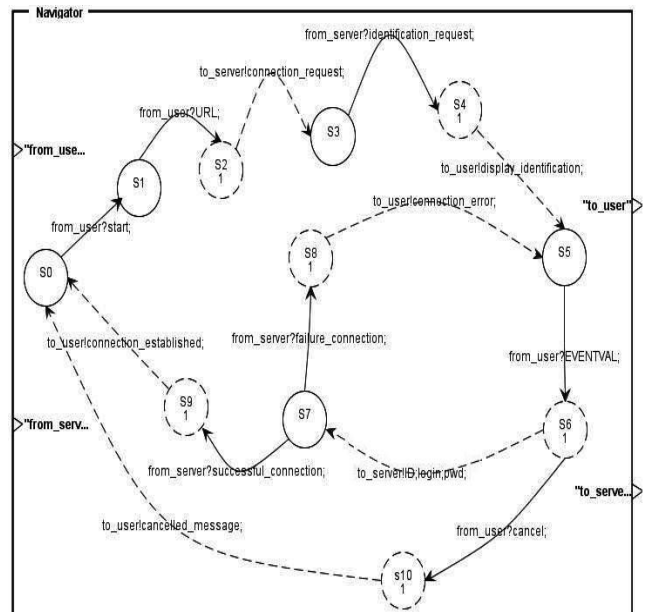


Figure 14: Global atomic GDEVs model of navigator

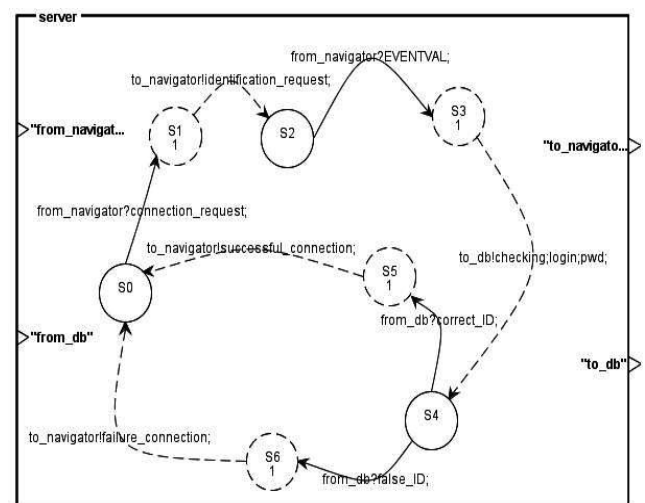


Figure 15: Global atomic GDEVs model of server

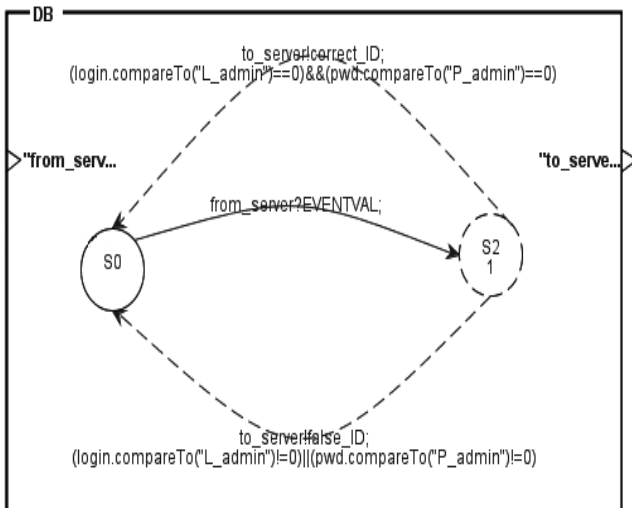


Figure 16: Global atomic GDEVS model of DB

When the Server sent message checking(login, pwd) to the database DB, this message is represented in the server GDEVS atomic model by an internal transition composed by the set of event “checking;login;pwd”:

- δ_{int} : to_DB! checking;login;pwd; and when the database DB receive this message, the latter is represented in the DB GDEVS atomic model by an external transition:
- δ_{ext} : from_server? EVENTVAL; The values of the message and its parameters are stored in variables in the state after the external transition as follows: msg=EVENTVAL[0]; login=EVENTVAL[1]; pwd=EVENTVAL[2];

After receiving the values of parameters, we check if they are correct or no. After the global atomic models for all objects of the system have been built, we construct the coupled model given in the figure 17 who describes the overall behaviour of the system.

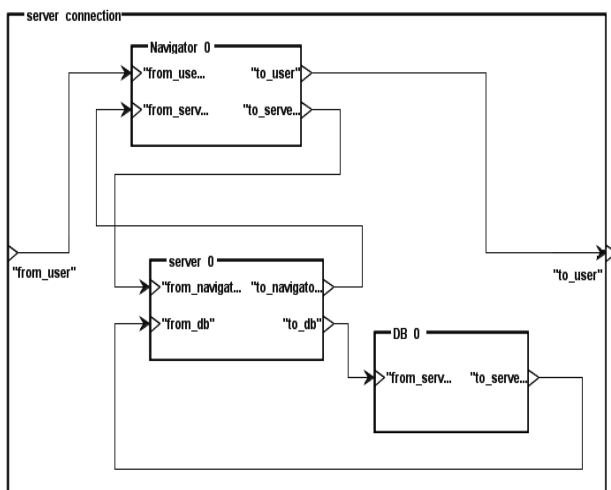


Figure 17: A coupled GDEVS model for server connection

To validate the specification of the behavior of the system obtained, we simulate the model results. For that we use the LSIS-DME tool. This simulator was developed by team members of our laboratory; it's composed of two parts: a model editor, and simulator. The editor is a module that graphically represents DEVS or GDEVS, atomic or coupled models. Once the models are created, they are saved in a library. Then, from a model and a set of data (Figure 18), the simulator provides the simulation results (Figure 19). The dataset is defined by the user who enters all external events supposed to occur during the simulation. For that, he entered for each event the port of the model on which the event will occur and the value of the event. During the simulation, we have to check that all events were treated.

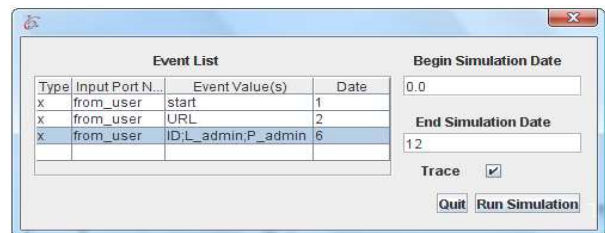


Figure 18: Example of scenario

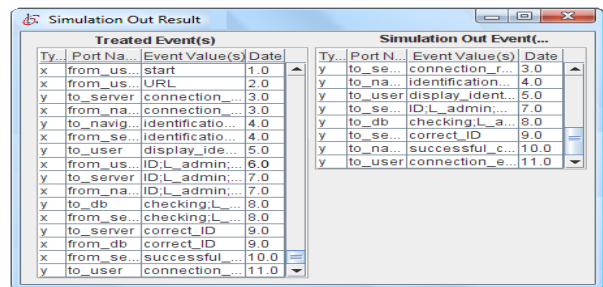


Figure 19: The simulation results

5.1 The structure of the system

Normally, we take three steps to develop a simulation GDEVS model: First, we specify the system with a minimum set of UML modeling elements to make the modeling process simple and efficient. We choose three UML diagrams: the Use Case (UC) diagram for requirements elicitation, the Class Diagram CD for specifying the structure of the system, and the Sequence Diagram SD for specifying the interaction between the system objects and for representing the system behavior. In domain analysis, we concentrate on the requirements that the modelers want to validate before real system is developed. We elicit the requirements with UCs, specify the inter-objects interactions with the constrained SDs, and describe the structure of the components with constrained CDs. A CD shows the static structure and the relationships between classes in the system (Eriksson et al. 2004). The relationships can be association, dependence, specialization, package, etc. However, we don't need all the relationships; we use only the association and composition relationships, which can represent the hierarchical structure of a system.

In Class diagrams, if an object O1 is defined like composition of many other objects O1.1... O1.n, then during the synthesis, the atomic models corresponding to the objects O1.1... O1.n will constitute a coupled model representing the total behavior of the object O1 (Figure 20) and who will belong to final model coupled GDEVS representing the total behavior of the system as in the following example.

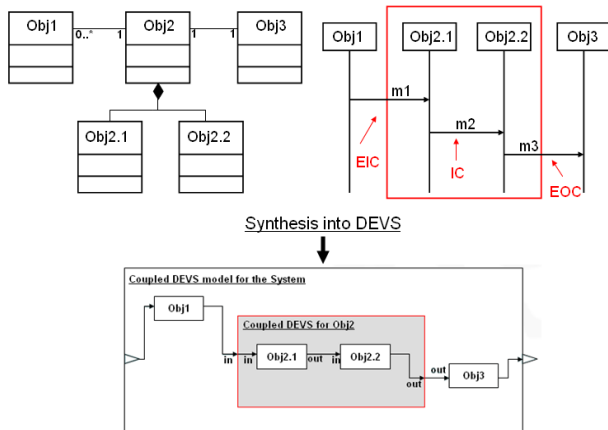


Figure 20: If an object is composed by an others objects in class diagram.

Then, after all atomic models are defined. We can draw information of the coupled model from atomic models and class diagram. This is because we represent class diagrams in a hierarchical form. For coupled modeling, we first identify components of a coupled model from class diagrams. Then, we identify external input/output coupling between the coupled model and components. Finally, we identify an internal coupling among the component models (Figure 17).

6 INTEGRATION OF FORMAL METHODS IN SYNTHESIS PROCESS

Although simulation is crucial for validating the behavior of dynamic systems, it can not cover all the possible cases of system evolution. In addition it doesn't allow for formal verification of the specification. In fact, it is unable to detect eventual conflicts in some requirements of the system like “ property 1: the system is *always in move*,..., property n: when the system *stops* the door has to be open ”, it is also unable to detect ambiguities like “ property 1: IF system_temperature>20 THEN the cooling system has to be activated, property n: IF system_temperature>40 THEN the fire alarm has to be released, as a result the system can not choose between activating cooling system and releasing fire alarm when system_temperature>40 ”, it is unable to detect incompleteness too, like “ property1: WHEN $0 \leq \text{car_speed} < 50$ THEN the electric engine is activated, property n: WHEN $50 < \text{car_speed} < 220$ THEN the thermal engine is activated, the question here which engine will work when the car_speed=50? ”. Therefore, we aim to integrate formal methods in our approach, to add credibility in simulation results and make the specification stronger. For that end, we use the approach

given in (Trojet et al. 2008) and (Trojet et al. 2009) which consists in transforming a DEVS model to a Z specification in order to reason formally on the resulting specification (applying mathematic techniques) using Z tools already available.

The following example illustrates the usefulness of the integration of FMs in our approach.

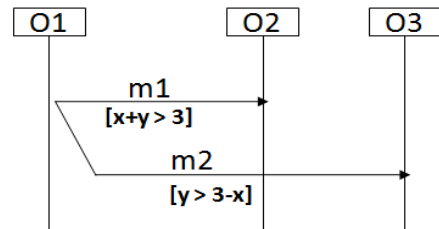


Figure 21: scenario representing ambiguities

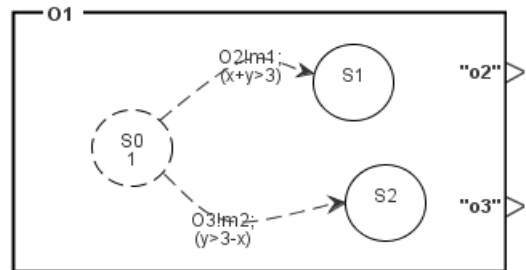


Figure 22: Atomic model DEVS of the scenario in figure 21

The Z specification equivalent to the atomic DEVS model is the following:

$PHASE ::= S0 \mid S1 \mid S2$

State

phase: PHASE

$x, y: \mathbb{N}$

Internal_transitions

$\Delta State$

$phase = S0 \wedge x + y > 3 \Rightarrow phase' = S1$

$phase = S0 \wedge y > 3 - x \Rightarrow phase' = S2$

Z tools permit to detect quickly the ambiguity which occurs in the two internal transitions of the DEVS model. In fact, the latter is blocked in the state S0 because conditions on transitions released from S0 are equivalents ($x+y>3 \Leftrightarrow y>3-x$), thus the model can not decide about next state. Therefore requirements must be redefined and the scenario has to be fixed.

7 CONCLUSION

State and interaction based modeling are two complementary approaches for the system specification. The former permits to capture the behavior of each system object, and the latter describes the

interoperability between all the objects. In this paper, we have shown the way to combine these approaches in order to validate all over the system behavior. The powerful side in our work is that we have derived the first approach from the second one under some rules we have defined. Therefore, the automatic extraction makes the integration process cost effective. In addition, our approach permits to check specification consistency over two levels: the first deals with the conformity of GDEVS model with scenarios, in fact, our proposal allows to check, via simulation, whether the global behavior represented by the GDEVS coupled model is similar or include the one described in the set of scenarios. The second level deals with the logical semantics of the specification; indeed, our proposal makes possible the formal verification of logic inconsistencies which can occur eventually in requirements or modeling phases (properties incoherence, missing assumptions, types errors, ambiguities, etc).

REFERENCES

- Alur, R., Etessami, K., Yannakakis, M., 2003. Inference of Message Sequence Charts, *IEEE Transactions on Software Engineering*, Vol. 29, No. 7, pp.623-633.
- Arjan, J., Mooij, N. Goga, and Judi, M.T Romijn., 2005. Non-local choice and beyond: Intricacies of MSC choice nodes. In *Maura Cerioli, editor, Fundamental Approaches to Software Engineering: 8th International Conference, FASE*, volume 3442 of *LNCS*, pages 273-288.
- Brian, L., Hans, E., 2004. UML 2 toolkit. *Wiley Publishing OMG press*.
- Damas, C., Lambeau, B., Lamsweerde A., 2006. Scenarios, Goals, and State Machines: a Win-Win Partnership for Model Synthesis. *14th ACM SIGSOFT International Symposium on Foundations of Software Engineering*, pp. 197-207. Portland, Oregon, USA.
- Damm, W., Harel, D., 2001. LSCs: Breathing life into message sequence charts. *Formal Methods in System Design*, 19(1):45--80.
- Eriksson H., Penker M., Lyons B., and Fado D., *UML 2 Toolkit*, Wiley Publishing, IN, 2004.
- Giambiasi N., Escude B., Ghosh S., 2000. GDEVS A Generalized Discrete Event Specification for Accurate Modeling of Dynamic Systems. *SCS Transactions*, Volume 17, 3, p.120-134.
- Hamri M. E.-A., Zacharewicz G., 2007. LSIS DME: An Environment for Modeling and Simulation of DEVS Specifications, in: *AIS-CMS International modeling and simulation multiconference*, pp. 55-60, Buenos Aires - Argentina.
- Harel D., Kugler H., Pnueli, A., 2005. Synthesis Revisited: Generating Statechart Models from Scenario-Based Requirements. *Formal Methods in Software and Systems Modeling*.
- Hélouët Loïc, Claude Jard, 2000. Conditions for synthesis of communicating automata from HMSCs, *5th International Workshop on Formal Methods for Industrial Critical Systems (FMICS)*, Berlin, 3-4 avril 2000.
- ITU 2000. Message Sequence Charts. Recommendation Z.120. *International Telecommunications Union. Telecommunication Standardisation Sector*.
- Letier, E., Kramer, J., Magee, J., Uchitel, S., 2005. Monitoring and Control in Scenario-Based Requirements Analysis. *International Conference on Software Engineering, Proceedings of the 27th international conference on Software engineering*.
- Liang H., Dingel J., Diskin Z., 2006. A comparative Survey of Scenario-based to State-based Model Synthesis Approaches, *SCESM'06 : International Workshop on Scenarios and State Machines: Models, Algorithms, and Tool*, Shanghai, China, pp.5-12.
- OMG 2005. UML 2.0 Specification. *Object Management Group. Available from: http://www.omg.org* [August 2005].
- OMG 2007. Unified Modeling Language: *Superstructure, v2.1.2*. Document Number: formal/2007-11-02, available specification <http://www.omg.org/spec/UML/2.1.2/Superstructure/PDF>.
- Sqali M. and Torres L., 2008. Modeling and Simulation of Behavioral Scenarios by Using Coupled DEVS Models. in: *The International Workshop on Modelling & Applied Simulation (MAS'08)*, Campora San Giovanni, Amantea (CS), Italy.
- Sqali M., Torres L. and Frydman, C., 2008. Synthesizing scenarios to DEVS models. *SpringSim08*. Ottawa, Canada.
- Trojet, M. W., A. Hamri, and C.Fydman. 2008. Logical analysis of DEVS models using Z. *Proceedings of International Simulation Multi-conference ISMc'08*.
- Trojet, M. W., C.Fydman and A. Hamri. 2008. Practical application of lightweight Z in DEVS framework. *Proceedings of Spring Simulation Multi-conference SpringSim'09*. San Diego, CA, USA.
- Woodcock J., and Davies J., 1996. Using Z: Specification, Refinement, and Proof. *Prentice-Hall*.
- Zeigler B., 1976. Theory of Modeling and Simulation. *Krieger Publishing Company*. 2nd Edition. NY.
- Zeigler B.P., Praehofer H., Kim T. G., 2000. Theory of Modeling and Simulation. *2nd Edition, Academic Press*, New York.
- Ziadi, T., Hélouët, L., Jézéquel, J., 2004. Revisiting Statechart Synthesis with an Algebraic Approach. *Proc. of 26th International Conference on Software Engineering (ICSE)*, IEEE Computer Society. p. 242-251. Edinburgh, May.

A GENERAL APPROACH TO HIGH FIDELITY MODELING, SIMULATION AND CONTROL OF TACTICAL ENTITIES AND IMPLEMENTATION IN A COMMERCIAL COMPUTER GENERATED FORCES TOOLKIT

Deniz Aldogan^(a), Mehmet Haklıdır^(b), Levent Hakkı Senyurek^(c), Seniha Koksall^(d), Omer Eroglu^(e), Cemil Akdemir^(f), Semuel Franko^(g), Isa Tasdelen^(h), Sevgi Akgun⁽ⁱ⁾

TUBITAK Marmara Research Center, Information Technologies Institute,
41470, Gebze-Kocaeli, TURKEY

^(a) deniz.aldogan@bte.mam.gov.tr, ^(b) mehmet.haklidir@bte.mam.gov.tr, ^(c) levent.senyurek@bte.mam.gov.tr,
^(d) seniha.koksall@bte.mam.gov.tr, ^(e) omer.eroglu@bte.mam.gov.tr, ^(f) cemil.akdemir@bte.mam.gov.tr,
^(g) semuel.franko@bte.mam.gov.tr, ^(h) isa.tasdelen@bte.mam.gov.tr, ⁽ⁱ⁾ sevgi.akgun@bte.mam.gov.tr

ABSTRACT

High-fidelity modelling for motion, response to physical stimuli and attack behaviours along with realistic control are vital requirements for training-critical military simulations. In addition, whenever simulation components are built on different infrastructures, maintaining coordination becomes a challenging problem. There are several COTS frameworks that facilitate development of computer generated forces (CGF), scenario management and run control for distributed simulations. However, they may not always afford to meet the requirements of virtual military training simulation systems. In this paper, we present modeling and simulation of surface vessels, submarines and rotary wing platforms with complex motion equations, several sensors, weapons and fuzzy logic controllers within a high-fidelity virtual military training simulation. Modules and libraries for high-fidelity entity simulation are integrated into component architecture of a commercial CGF simulation engine, namely VR-Forces. An original control architecture, which is able to manage all kinds of simulation components with different infrastructures, has also been realized.

Keywords: Computer Generated Forces (CGF), Military Training Simulation, HLA, RTI

1. INTRODUCTION

The requirement for building virtual military training simulations can be mostly traced to the fact that military field exercises are generally too costly to be used generously for staff training on military instruments. Such military simulations generally utilize a distributed simulation infrastructure since they mostly require reduced execution time, geographical distribution, fault tolerance and

integration of simulators that execute on different hardware platforms (Fujimoto 2000).

For training-critical military simulations, one of the most crucial tasks that can be assigned to a simulation component is the modelling and simulation of army forces. In order to meet the needs of high-fidelity military simulations, both realistic models and realistic controls should be employed for realizing the behaviour of computer generated forces (CGF). Besides CGF capabilities, tactical simulations mostly require a scenario preparation application. Another equally important necessity in such systems is the achievement of cooperation between simulators built on different infrastructures in the simulation. This integration may become challenging for the control architecture of the system.

A COTS toolkit can be utilized for developing a complex military distributed simulation in order to save time and effort. Many commercial tools are readily available for providing the fundamental CGF and scenario management capabilities. However, they may not provide a control architecture that is able to manage simulators of different infrastructures. In addition, they may not satisfy the necessities of a complex military system with high-fidelity platforms that exhibit advanced dynamic behaviours and other capabilities such as land or entity avoidance.

This study aims to provide an overall view of an on-going simulation system, which embodies commercial CGF toolkit-integrated simulation of a wide range of high-fidelity platforms. These platforms use complex motion equations and realistic controllers in which land avoidance behaviours are integrated into fuzzy controllers for course keeping and speed control of entities. The simulation system under study necessitates the integration of commercial framework independent

simulation applications that are mainly responsible for algorithmic behaviours. These simulators are implemented by using an original infrastructure to avoid the cost of using a commercial tool and to maintain reusability. Therefore, a simulation that embodies simulators using both a commercial infrastructure and a different original infrastructure is to be realized.

As a test application to be used for implementing the CGF unit and a front-end unit, which is responsible for supplying GUI, scenario management and run control operations, belonging to the system, VR-Forces has been chosen due to the fact that it facilitates easy customization and extension of its capabilities. VR-Forces is a flexible software that is produced as a solution for military simulation purposes. It uses RTI (Run-Time Infrastructure) for providing communication and coordination services to the simulation components. Its simulation components are of two types: front-end and back-end applications. The back-end application is where the actual CGF simulation takes place whereas the remote front-end GUI controls the back-end(s) existing in the whole simulation. VR-Link is an isolation layer which has been developed for enabling VR-Forces based applications to be used in DIS or HLA compliant simulations. It keeps the user away from the burden of architectural details (Figure 1). For better understanding the reasons for the usage of HLA compliant simulations and currently offered HLA services, Duman et al. (2003) can be referred.

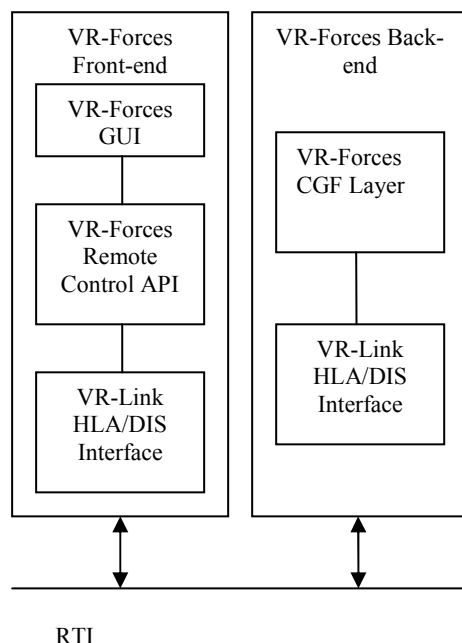


Figure 1: VR-Forces Architecture

The GUI application provides features for scenario management and simulation execution control. As for the scenario management, it enables

choosing a battle field for the current tactical scenario, addition, deletion and modification of predefined platforms, task assignments and task planning for the entities in the scenario. It also achieves overall scenario run control by its embedded remote control mechanism via sending messages for starting, stopping, loading or saving the scenario on the back-ends.

Back-end applications are responsible for modelling and simulating the CGF, managing property settings and task executions of entities according to the user commands sent from GUI, broadcasting each of the CGF component attributes to the simulation, receiving the attributes of CGF components on other simulation assemblies and responding to the run control messages delivered by the remote control on the GUI application. The details about the inner workings of VR-Forces architecture can be sought in VR-Forces Developer's Guide (2006) or "VR-Forces The Complete Simulation Toolkit" article on the website belonging to the company named "MAK Technologies". It should hereby be noted that the entities simulated by VR-Forces have simple motion equations with environmental parameters that are not of appropriate fidelity to develop realistic movement models.

The software modules for the motion models, sensors, weapons and the fuzzy controllers belonging to platforms have been implemented in the C++ programming language and have been integrated to the component architecture of VR-Forces CGF application (VR-Forces back-end) as composite objects.

The rest of the paper is divided into 5 sections. In Section 2, integration of motion models into the entities in the simulation system is given. Section 3 is dedicated to the implementation of fuzzy logic controllers and maintenance of land avoidance. Section 4 explains the addition of sensor and weapon models to the entities. A control architecture built from scratch to coordinate both commercial toolkit dependent and independent simulators is in Section 5. Finally, possible extensions and conclusions are given in Section 6.

2. INTEGRATION OF HIGH-FIDELITY MOTION MODELS INTO THE SIMULATION SYSTEM

In this study, movement models that take into account the environmental conditions (wave, current, wind, season, day and night difference) and hydrodynamic forces have been developed for surface, submarine and rotary wing platforms each with 6 degrees of freedom. The motion of all platforms are considered in 6 degrees of freedom since six independent coordinates are necessary to calculate state information of a rigid body.

Modelling details such as coordinate frames, motion equations, forces and moments acting on

each type of platforms as well as simulation results are described in the following papers: the type of platform under study is surface vessels in Haklidir, Aldogan and Tasdelen (2008), rotary wing platforms in Franko, Koksals and Haklidir (2009) and submarines in Haklidir, Guven, Eroglu, Aldogan and Tasdelen (2009). A software library is implemented for the motion of each type of platform to maintain modularity and reusability.

The integration of the software modules for motion models into the back-end application, i.e. the current CGF simulation engine, is realized by the usage of VR-Forces component architecture. VR-Forces back-end(s) simulate tactical object behaviours by utilizing a component architecture. According to this architecture, there are three different types of components that an object may possess: sensors, controllers and actuators. Sensors help entities to gain information about their surrounding environment. Controllers receive task commands from GUI and direct the entity for the completion of the assigned tasks. To illustrate, when an entity is ordered to follow a given route defined on the current scenario, the controller specialized for this task may provide the rudder angle or shaft r.p.m.(revolutions per minute) to the actuator component, which actually performs the entity motion by applying the relevant motion models and updating the entity's state information such as position, speed, etc. according to the outcomes of the models used. There are other actuator types such as weapon or damage actuators. An object may have several components of same or different types. For an object, three component lists are maintained, i.e. list of sensor components, etc. The component manager belonging to the entity ticks each of the components while preserving this order: sensors, controllers and actuators.

In the military simulation system that is introduced in this study, the movements of the entities must be governed by high-fidelity motion models. Each of these motion models have been developed as VR-Forces independent software modules. These modules are then integrated into the VR-Forces component architecture. This is achieved by following the steps explained below for each different entity category, i.e. surface vessels, submersible vessels and rotary-wing entities:

- A new actuator class has been derived from the original VR-Forces component that is specific to the entity domain, i.e. an actuator for surface vessels, etc.
- A software module that calculates the next state of the entity given the current state values and commanded task related parameters has been implemented.
- The software module for realizing the entity behaviour is defined as a member

variable to the corresponding actuator class.

- The main loop of the actuator class is overridden so that it updates the platform's state information such as position, speed, etc. by utilizing the software module in each simulation frame time.

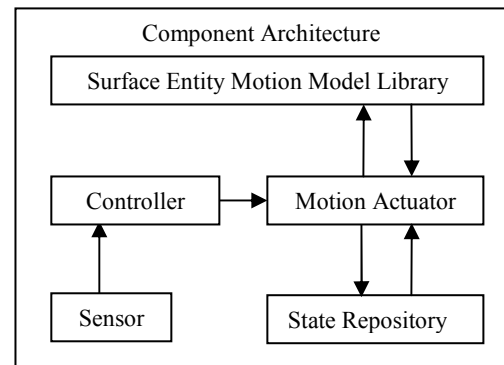


Figure 2: Integration of Surface Entity Specific Motion Model Library into the VR-Forces Component Architecture

Figure 2 displays the integration of a high-fidelity model library, motion model library for surface entities in this case, into the component architecture of VR-Forces CGF application. The data flow between the components has been displayed as arrows. All the components in the figure except from the model library have been derived from original VR-Forces components to adapt the model library.

3. INTEGRATION OF FUZZY CONTROLLERS AND LAND AVOIDERS FOR GUIDING THE ENTITY

Realistic control is a crucial necessity to simulate entities of high-fidelity in military simulation systems. In this study, commercial toolkit independent controllers are used to simulate quartermasters of the surface platforms. A flexible fuzzy logic library (FLL) capable of simulating human expert behavior has been implemented in the C++ programming language. Fuzzy logic controllers (FLC), which are in fact heading and speed controllers that utilize FLL for their calculations, are implemented in conjunction with land avoidance calculations.

A multi-input-single-output (MISO) PD type FLC for controlling the direction and a MISO PI type FLC for controlling the speed of the surface vessel have been produced. Big Bang – Big Crunch (BB-BC) optimization algorithm, proposed by Erol and Eksin (2006), has been utilized to sample some reference course/speed values in the physically

allowable range and to find corresponding scaling factors. The error between the desired and the actual heading and speed values are used as controller inputs along with their integrals and derivatives. The inputs and the outputs are defined by seven linguistic variables over the specified range, namely, NL (negative large), NM (negative medium), NS (negative small), Z (zero), PS (positive small), PM (positive medium) and PL (positive large). For input representation, symmetric triangles have been used as membership functions (MFs), whereas singleton membership functions have been chosen for the output MFs. The rule base for FLC consists of 49 rules, each of which are displayed in Table 1. To illustrate, when the input values are PS for Δe and PS for e , the output becomes a PM value.

Table 1. The Rule Base for Fuzzy Logic Heading and Speed Controllers

$\Delta e/e$	NL	NM	NS	Z	PS	PM	PL
PL	Z	PS	PM	PL	PL	PL	PL
PM	NS	Z	PS	PM	PL	PL	PL
PS	NM	NS	Z	PS	PM	PL	PL
Z	NL	NM	NS	Z	PS	PM	PL
NS	NL	NL	NM	NS	Z	PS	PM
NM	NL	NL	NL	NM	NS	Z	PS
NL	NL	NL	NL	NL	NM	NS	Z

The implemented fuzzy logic controllers have been compared with classical PD controllers even in extreme cases. Results show that the designed fuzzy logic controllers are superior to classical PD controllers in all cases. A more detailed explanation as well as simulation results for FLC's can be found in Senyurek et al. (2008).

Below, the integration of originally developed FLCs into the component architecture of the commercial toolkit, specifically into the task controller component in the VR-Forces back-end architecture, is explained by describing a sample flow of control for the surface vessels.

When a task such as moving to a waypoint or moving along a route has been commanded by the user from GUI application, the entity's task controller, which is the VR-Forces component responsible for adjusting the bearing and speed parameters for the current task, receives a message related to the assigned task through the VR-Forces messaging interfaces. The obtained message includes parameters related to the entity's assigned task. These parameters are mainly commanded point/bearing, commanded speed and/or commanded task duration. Ordered bearing value with respect to the current commanded point is calculated unless a commanded bearing value is specified directly during task assignment. The parameters are then transferred to FLCs, i.e rudder and shaft controllers for the surface vessels, which

are owned by the entity's commercial toolkit based task controller component. In each simulation frame, FLCs supply FLL with the error between the desired bearing/speed and the actual heading/speed values along with their integrals and derivatives. By the way, heading refers to the direction that the entity is headed, while bearing means the direction that the entity ought to be headed.

The outcomes of FLL calculations are rudder angle and shaft speed for the entity movement. These values are stored to be used by the actuator component. The actuator component takes these values and feeds them to the motion module, which is owned and managed by the actuator, to calculate the entity's next position. These steps are repeated until the task is accomplished.

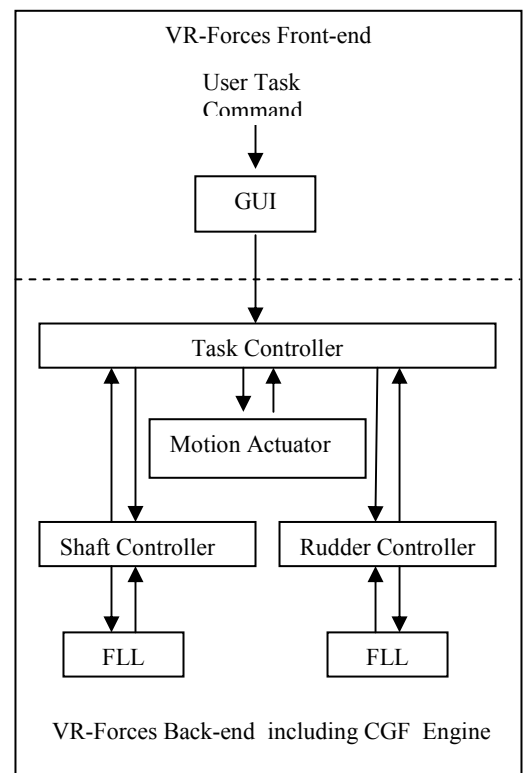


Figure 3. Integration FLLs into the VR-Forces Task Controller Component

Figure 3 displays the integration of FLCs and FLLs into the component architecture of VR-Forces CGF application. The data flow between the components has been displayed in the form of arrows.

The task control process mentioned above deviates from its normal flow whenever there are land points within a certain range around the entity. Then, a land avoidance vector can be calculated for the surface entity. This vector is calculated by using the points of land area on the scenario map within a predefined proximity to the entity. The direction from the gravity centre of the relevant land points to the entity gives the direction

of the land avoidance vector for the current simulation frame time. The magnitude of the current land avoidance vector is proportional to the distance from the entity to the calculated centre of gravity. For now, a land avoidance vector for an entity with an on-going task is calculated at each simulation frame. Here, an optimization process is vital so that land avoidance vectors are calculated less often to relieve the resulting time complexity.

Land avoidance vectors are used to keep the surface entity away from land areas. The x, y components of the land vectors are combined with ordered bearing and speed parameters of the task at hand. The outcome is the new ordered bearing and speed values that take into account the land avoidance vectors.

In Figure 4, a sample gravity center G , the corresponding land avoidance vector a and its projection on the entity a' , the surface entity S , its ordered bearing direction b and finally the combined ordered bearing vector c are given. Owing to the calculated land avoidance vector, the entity is able to carry out the commanded task by moving around the land points.

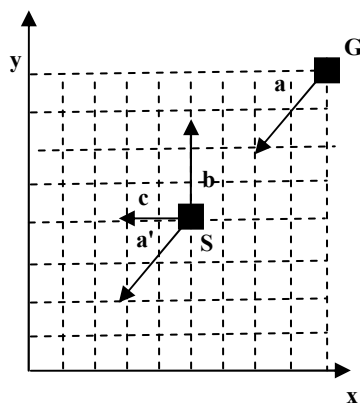


Figure 4. The Illustration for the Combination of a Sample Land Avoidance Vector to the Entity's Ordered Bearing Vector

The flow of control explained above continues by supplying the modified ordered bearing and speed parameters to the FLL.

4. INTEGRATION OF SENSOR AND TORPEDO LIBRARIES INTO THE PLATFORMS

Sensor components of a platform provide models of the simulated environment, which are then used by the controller components to make decisions and perform tasks.

Currently, there are three types of sensors that an entity in our system may have. These are passive sonar, active sonar and radar components.

Natural and traffic noises belonging to the acoustic communication along with entity's own noises are produced by a noise generating

simulation component according to entity's propeller revolutions per minute (rpm). These data are sent in the form of object updates via RTI to the CGF simulation component. This component possesses lists that maintain such remote objects.

Besides, there exists an acoustic transmission simulation component that is responsible for supplying the necessary transmissions in the form of interaction messages for the sensor components to work.

Each sensor model is implemented as a software module. Developing every advanced feature to be integrated into the entity behaviours as commercial toolkit independent software modules has been adopted throughout the realization of our system to maintain modularity and reusability.

A VR-Forces component has been derived for each sensor type. The corresponding sensor library has been integrated into the component as a composite object. This policy closely resembles the method that is used to embed motion models and fuzzy control.

5. CONTROL ARCHITECTURE FOR THE INTEGRATION OF DIFFERENT SIMULATION SYSTEMS

Maintaining coordination for scenario management and simulation execution control behaviours become a critical problem whenever simulation components which are devoted to perform different responsibilities are built on different infrastructures in the same simulation. An original infrastructure in addition to a commercial toolkit infrastructure, namely VR-Forces, is used in the current simulation system. The reason for this utilization of an original infrastructure can be mostly traced to two main facts. Firstly, requirements for the simulation of algorithmic behaviours such as noise generation or acoustic transmission are not fully satisfied by the commercial toolkit at hand. Moreover, increasing the licence costs of the commercial toolkit can be avoided while maintaining reusability.

Two different solutions for the problem of integrating different simulation infrastructures can be stated: modifying the remote control mechanism of the commercial application(s) as displayed in figure 4 or developing an original control application from scratch, which is illustrated in figure 5. The second solution has been adopted in the current study since it provides isolation from the commercial framework, modularity and reusability. The control architecture developed for the current hybrid infrastructure is HLA and RPR-FOM (Real-Time Platform-Level Reference Federation Object Model) compliant. That means communication between different simulation components is performed by sending/receiving messages that are in the form of object and

interaction classes that are defined in the file named “Federation Object Model (FOM)” (Figure 5).

If the whole simulation under study, i.e. all the simulation components in the simulation, had been developed by using only VR-Forces infrastructure, the remote control ability on the VR-Forces GUI would be sufficient for run control. All of the scenario and run control messages of VR-Forces remote control are sent by using the same interaction class, namely the Data Interaction, defined in RPR-FOM. Owing to a VR-Forces specific encoding/decoding layer, this interaction is decoded on the receiving VR-Forces back-end. Adopting the same process for the VR-Forces independent simulation components in the current system has two main disadvantages. In addition to the burden of integrating a VR-Forces dependent encryption layer to all the VR-Forces independent simulation components, the denseness of messaging traffic would be highly raised since all the simulation components would receive all the messages sent by the remote controller whether they were relevant or not. This type of architecture is illustrated in Figure 4.

As a solution, a new simulation component built by the original infrastructure has been assigned as the scenario and run control manager. Therefore, the control of execution for the whole simulation has been isolated from the VR-Forces remote control mechanism. New interaction classes have been derived and used in the communication between the VR-Forces GUI and the new simulation controller for receiving user commands.

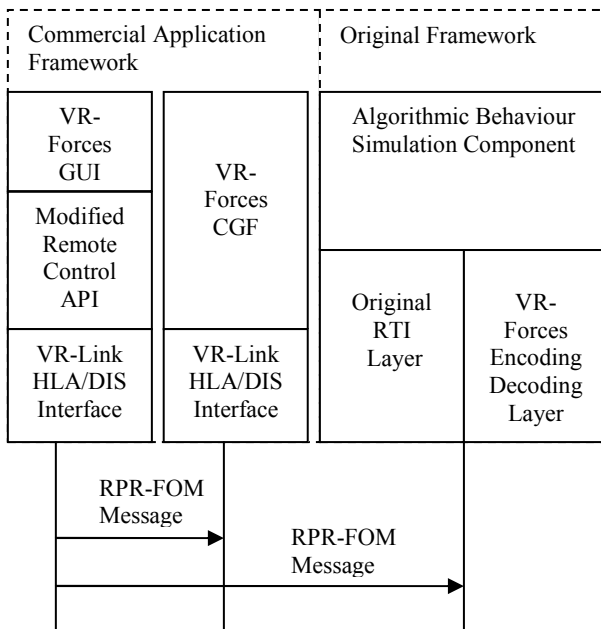


Figure 5. Display of Message Flow on the System Developed by Modification of VR-Forces Remote Control API

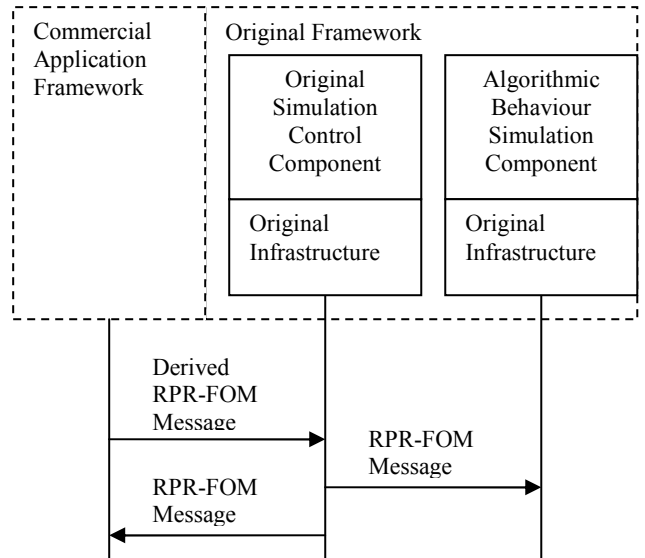


Figure 6. Display of Message Flow where an Original Simulation Control Component is utilized

Upon receiving a derived interaction corresponding to a user command such as “load/save scenario” or “start/stop simulation”, the simulation controller sends the corresponding standard RPR-FOM interaction messages to the simulation components with the original infrastructure. In this way, the execution of the whole simulation is controlled.

6. EXPERIMENTAL RESULTS

6.1 Results on Surface Entities

Surface platforms are tested in a scenario in which a surface platform makes four zig-zag maneuvers of 45° with a velocity of 8 m/s. The route information regarding this task is inputted by defining four waypoints as shown on the VR-Forces graphical user interface (Figure 7).

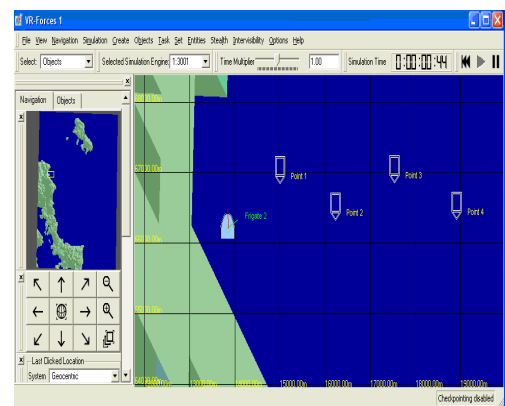
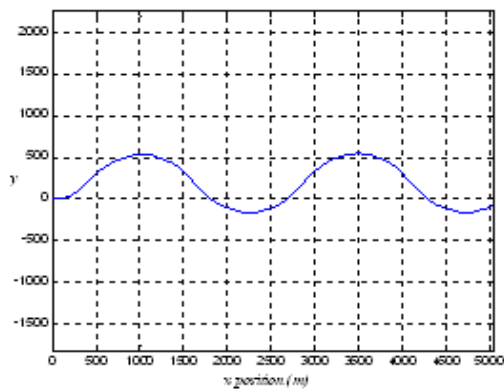
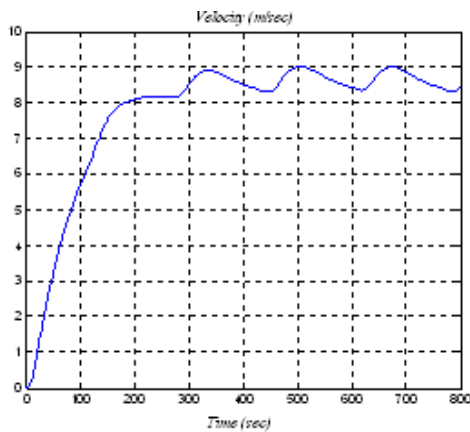


Figure 7. The Route Defined for the Platform

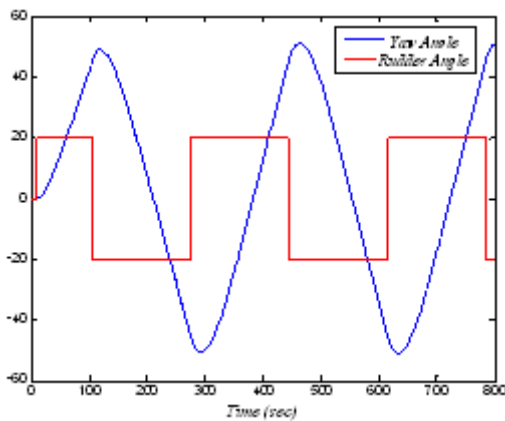
The results below have been produced after running the simulation for 800 seconds.



(a)



(b)



(c)

Figure 8. (a) Change of Location for the Surface Platform (b) Change of Velocity for the Surface Platform (c) Changes in the Yaw Angle and Rudder Angle of the Surface Platform

In this application, which is known as the zig zag test of Kempf in the literature (Kempf 1932), the initial speed of the platform has been given as 0. The platform is ordered to move to the specified waypoints one by one by increasing its velocity up to 8 m/s. It takes the platform 96 seconds to reach

to the first point. The first loop is accomplished in approximately 295 seconds. The results are acceptable for the movement behaviors that are supposed to be realized by a large platform and satisfactory in terms of simulation. Detailed discussion for simulation results can be obtained from the study Haklidir, Aldogan and Tasdelen (2008).

6.2 Results on Submarine Entities

6.2.1 Route Tracking

In this application, it is aimed to realize the submarine tracking on a non – equilateral pentagon route with 10 m/s cruise speed. Route information is defined via VR-Forces graphical user interface which can be seen by the red lines in Figure 9. The blue symbol to the left of the red route points refers to the submarine entity.

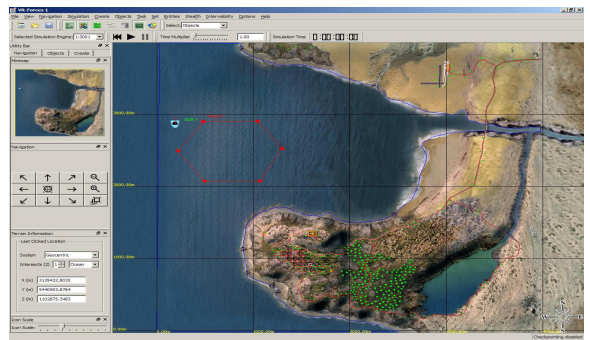
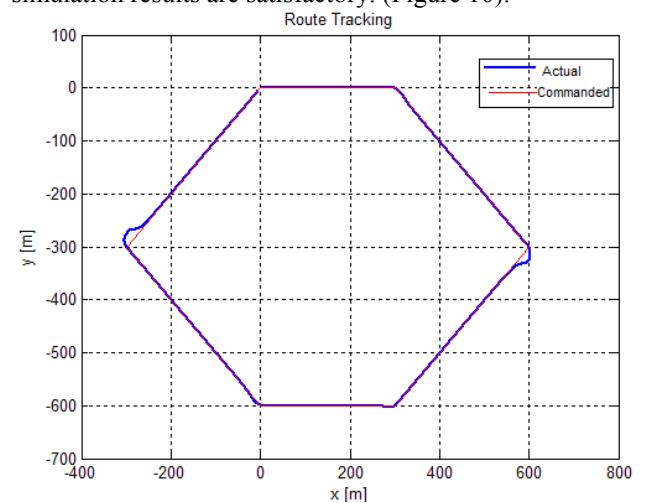
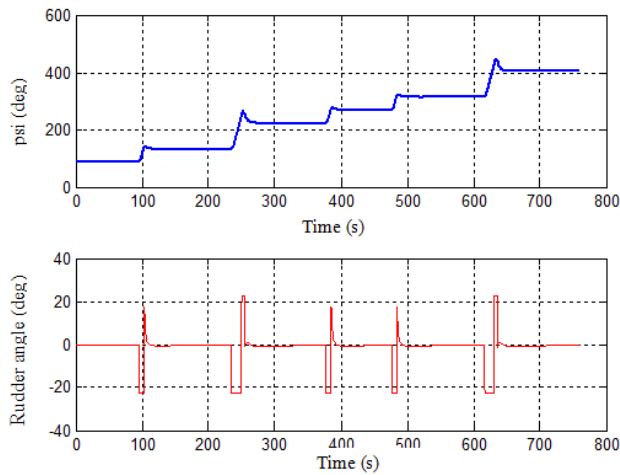


Figure 9. Display of the Route Defined by Red Points (and lines in between) on the VR Forces Graphical User Interface.

Simulation time is set to 800 seconds and the following results are obtained. In the light of the below results, it is clearly said that the model follows desired route successfully and the simulation results are satisfactory. (Figure 10).



(a)



(b)

Figure 10. (a) Submarine Route Tracking (b) Submarine Yaw and Rudder angles

6.2.2 Depth Control

To analyze the effectiveness of the submarine depth controller, firstly, the submarine entity is ordered to accomplish the following task. With a cruise speed of 10 knot (5 m/s), it initially dives to periscope level (14 m) and then down to 50 m. Simulation results are summarized as depth values and rudder angle differences (Figure 11 and 12 respectively).

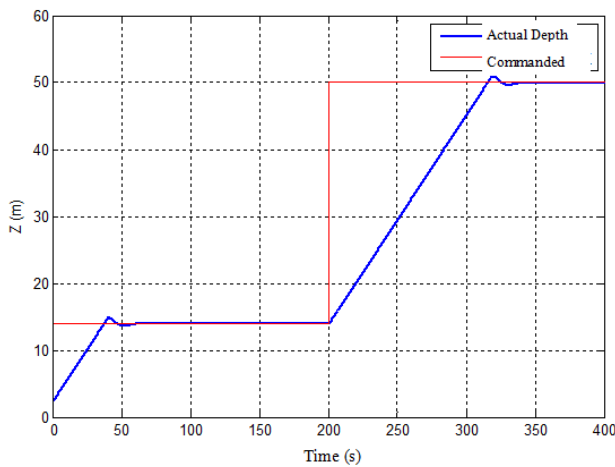


Figure 11. Submarine Depth Tracking

In this simulation, the submarine which hovers at 2.5 m dives to desired depths (14 m and then 50 m) in 42 seconds and 120 seconds respectively. Simulation results are acceptable according to dynamics of a submarine and they are satisfactory with respect to simulation performance, as well. Besides, the control signals are on an admissible level for realization. Detailed discussion for simulation results can be obtained from the study Haklidir et al. (2009).

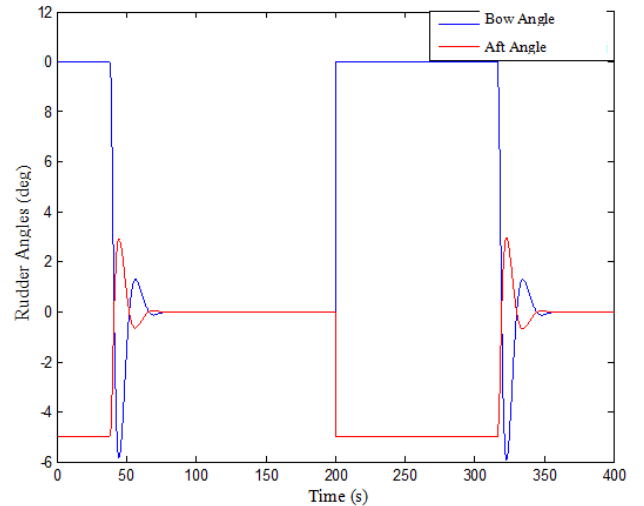


Figure 12. Submarine Bow and Stern Rudder Angles

6.3 Results on Rotary Wing Entities

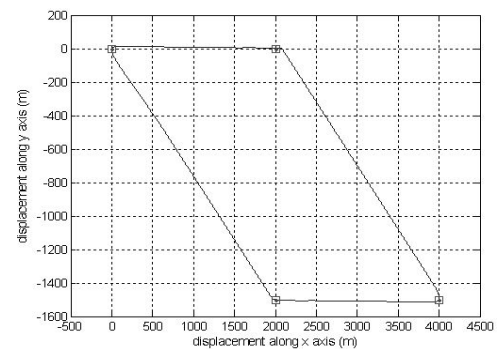
For testing the controllers' performance, the following scenarios were formed. In the first scenario, the user orders the helicopter to move to four waypoints one after another. In the latter one, a route has been established and the entity has been commanded to move along this route.

6.3.1 Movement Through Waypoints

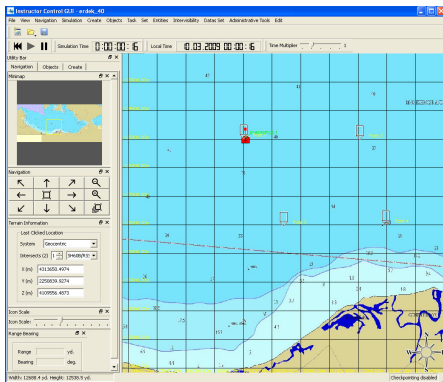
For testing the controller, a scenario that contains a waypoint has been formed. Initial point is set as $[x \ z \ y] = [0 \ 0 \ 0]$ meters. Commanded composite velocity (of u and v) is 20 m/sec. Four waypoints were selected as follows:

- Waypoint 1= $[2000 \ -300 \ -1500]$
- Waypoint 2= $[4000 \ -300 \ -1500]$
- Waypoint 3= $[2000 \ -300 \ 0]$
- Waypoint 4= $[0 \ -300 \ 0]$

In the figure below, the small circles on the corners of the quadrilateral indicate the waypoints, while lines that denote the edges es show the actual trajectory of the helicopter.



Figures 13: 2D View of Rotary Wing Entity Motion through Waypoints



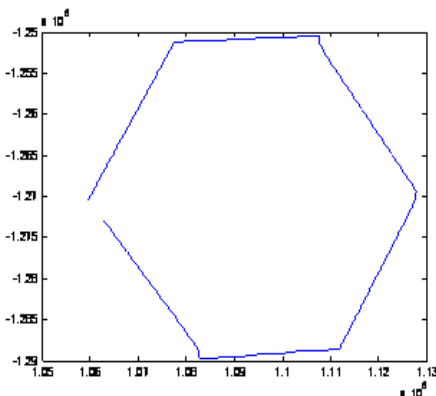
Figures 14: View of Rotary Wing Entity Motion through Waypoints in VR-Forces GUI

6.3.2 Movement Along Route

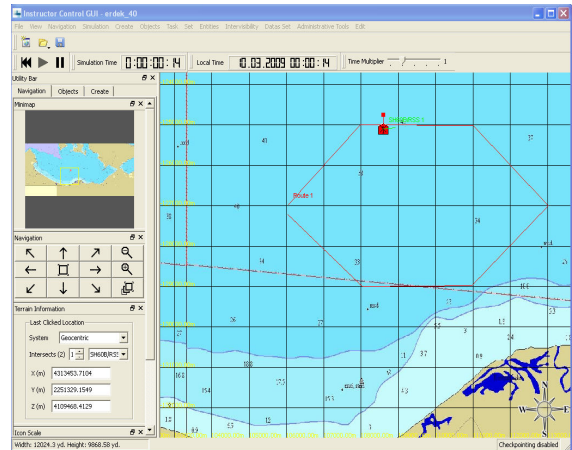
Initial point is set as $[x \ z \ y] = [0 \ 0 \ 0]$ meters. A route consists of seven points was selected as follows:

- Point 1= $[-1282482 \ 0 \ 1055818]$
- Point 2= $[-1271867 \ 0 \ 1059178]$
- Point 3= $[-1251208 \ 0 \ 1077556]$
- Point 4= $[-1250465 \ 0 \ 1107525]$
- Point 5= $[-1288555 \ 0 \ 1111868]$
- Point 6= $[-1289796 \ 0 \ 1082805]$
- Point 7= $[-1272898 \ 0 \ 1062745]$

Composite velocity was the same with that of the previous scenario. The result of the simulation can be observed in the following figure. Detailed discussion for simulation results can be obtained from the study Franko, Koksai and Haklidir (2009).



Figures 15: 2D View of Motion through Route



Figures 16: View of Motion through Route in VR-Forces

6.3 Fuzzy Library Simulation Results

Figure 17 and Figure 18 show the change of the heading of the ship with respect to time for two extreme cases. The Fuzzy Logic heading controller is compared with manually tuned PD controller in which classical Ziegler-Nichols method is utilized as a starting parameter set. In both cases, the initial heading angle is considered as zero. In the first case (Figure 17), the initial shaft speed of the vessel is at maximum level (240 rpm) and the ship is forced to a 5° (small) rotation, while in the second case the initial shaft speed is 70 rpm (resulting a slow speed value of about 5 knots) and the desired rotation is 180° (maximum possible change). The result obtained by the Fuzzy controller is represented by solid lines, while the dashed lines are the results of the classical PD controller.

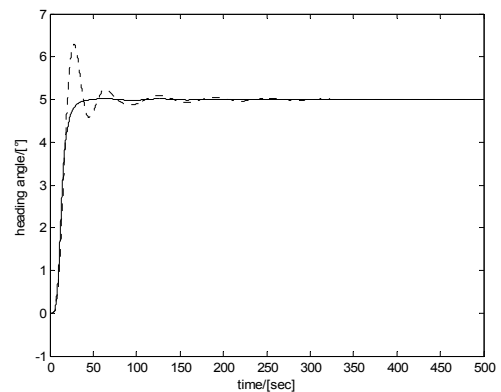


Figure 17. Change of Heading of the Ship in time, Controlled by a Fuzzy Logic and a Classical PD Controller (dashed line) in a High Speed - Small Rotation Case

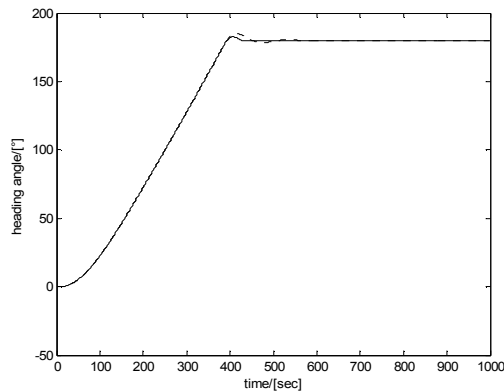


Figure 18. Change of Heading of the Ship in time, Controlled by a Fuzzy Logic and a Classical PD Controller (dashed line) in a Low Speed - Large Rotation Case

Results show that in both extreme cases, the designed fuzzy logic rotation controller is superior to classical PD control even in auto steering only. So it actually seems to be a good choice to use fuzzy logic for simulating quartermaster behavior, since FLC's are also superior for designing expert systems to simulate human operators.

Results also approve our choice of using two MISO fuzzy controllers for controlling speed and heading distinctly. Change in speed doesn't have a considerable effect on heading control. Detailed discussion for simulation results can be obtained from the study Senyurek et al. (2008).

7. CONCLUSIONS

This paper introduces an on-going study on a virtual military training system which uses high fidelity motion models ranging from 4 to 6 degrees of freedom, several sensor models, a torpedo model and fuzzy logic controllers that mimic captain control strategies for realizing behaviours of entities. The whole design is integrated into the component architecture of VR-Forces, which provides a framework for developing Computer Generated Forces (CGF) applications. In addition, commercial toolkit independent simulation components that specialize on algorithmic behaviours are integrated into the commercial toolkit based CGF and GUI applications via developing an original control architecture. The overall integration has led the surface, submarine and rotary-wing platforms of the VR-Forces application to reach a high-fidelity level that is necessary for training-critical military simulations.

REFERENCES

Duman, I., Savaşan, H., Şahin, I., 2003. Bir Modelleme ve Simülasyon Standardı olarak HLA ve Simülasyon Uygulamalarında Kullanımı, (In Turkish), *Turkish Navy Research Center Command Publications*.

- Erol, O. K., Eksin, I., 2006. A new optimization www.method: Big Bang–Big Crunch, *Advances in Engineering Software*, 37, 106–111.
- Franco, S., Koksall, S., Haklıdır, M., 2009. Modeling, Simulation And Control Of Rotary Wing Platforms In A Computer Generated Forces Toolkit, *The Industrial Simulation Conference ISC 2009*, Loughborough, United Kingdom, 1-3 June 2009
- Fujimoto, R. M., 2000. *Parallel and Distributed Simulation Systems*, Wiley Series on Parallel and Distributed Computing, pp. 4-5.
- Haklıdır, M., Aldogan, D., Tasdelen, I., 2008. High Fidelity Modeling and Simulation of Surface Platforms in a Computer Generated Forces Toolkit, *SUMMERSIM 2008*, Edinburgh, Scotland.
- Haklıdır, M., Guven, A. F., Eroglu, O., Aldogan, D., Tasdelen, I., 2009. High Fidelity Modeling and Simulation of Submarine in a Commercial Computer Generated Forces Toolkit, *The Society for Modeling & Simulation International [SCS] 2009 Summer Computer Simulation Conference (SCSC'09)*, Istanbul, Turkey, 13-16 July 2009.
- Kempf, G., 1932. Measurements of the Propulsive and Structural Characteristics of Ships, *Transactions of SNAME*, 40, pp. 42-57.
- Senyurek, L., Koksall, S., Genc, H.M., Aldogan, D., Haklıdır, M., 2008. Implementation Of Fuzzy Control For Surface Platforms In A Computer Generated Forces Toolkit, *International Conference on Computational Intelligence for Modelling, Control and Automation (CIMCA08)*, Vienna, Austria, 10-12 December.
- VR-Forces Developer's Guide*, MAK Technologies, 2006. Chapter 2, The VR-Forces Simulation Engine, Chapter 5, Creating New Components, Chapter 10, The VR-Forces Remote Control API.
- VR-Forces The Complete Simulation Toolkit*. MAK Technologies. Available from <http://www.mak.com/products/vrforces.php> [accessed 21 June 2009]

JAVA FOR PARALLEL DISCRETE EVENT SIMULATION: A SURVEY

I. Castilla^(a), R. Aguilar^(b)

^{(a)(b)}Department of Systems Engineering and Automation, and Computer Architecture. University of La Laguna. Spain

^(a)ivan@isaatc.ull.es, ^(b)raguilar@ull.es

ABSTRACT

Since the early 90s, when it was first released, Java has become one of the most widespread programming languages. Discrete Event Simulation and also Parallel Discrete Event Simulation have attracted more and more projects which are Java-based. This paper presents a brief survey on the tools and facilities that make Java such an attractive option for parallel simulation developers. Nevertheless, several drawbacks and lacks of the language are also exposed.

Keywords: Parallel Discrete Event Simulation, Java, Survey.

1. INTRODUCTION

Sometimes, modeling and simulation practitioners from the scientific community push into the background the computational model. Undoubtedly, the modeling and simulation methodology includes several steps which (should) demand a higher time investment, such as the conceptual model development and the validation and verification process. However, implementing a computational model is the only way to get real applications and results.

Choosing the right tool for developing a simulation model is far from being a trivial task. Hundreds of commercial tools, libraries, open-source projects are all around with different and appealing characteristics. Even more, an ad-hoc solution using a high level language is a tempting election when none of the tools looks close enough to our specific problem.

When surveying the literature on implementations of sequential and parallel Discrete Event Simulators (DES), Java appears as a common choice among developers. Some of the reasons stated for using this language are object orientation, portability, built-in thread support, automatic handling of memory, and network awareness.

This paper does not intend to be a flattering Java pamphlet but to expose and review the arguments that support the use of this programming language for the implementation of Parallel Discrete Event Simulators (PDES). Furthermore, several criticisms are introduced that draw the pros and cons of the language.

Section 2 reviews the main concerns about PDES. Section 3 and 4 examine some of the most important Java features, devoting special attention to Java

performance. Section 5 reviews some distributed Java libraries not necessarily related to simulation. The next section goes deeper into Java-based PDES, whereas section 7 looks at the future and enumerates the expected updates for the language. Finally, some conclusions are drawn.

2. A REVIEW ON PARALLEL DISCRETE EVENT SIMULATION APPROACHES

There is no consensus on the taxonomy used to classify the different approaches to exploit parallelism in Discrete Event Simulation. The list presented here does not pretend to be an exhaustive nor formal classification, but to cover most of research interest fields.

2.1. Dedicated Execution

Different processors execute specific sequential simulation functions such as random number generation, visualization, event list management and statistic collection. The management of the event list usually becomes the major bottleneck for this approach and drastically limits the amount of parallelism that can be exploited.

The classical reference for this approach is (Comfort 1984).

2.2. Hierarchical Decomposition

An event is decomposed into sub-events which are executed concurrently. Performance improvement by using this approach is strongly dependent on the model and the decomposition process is far from being trivial.

Concepcion (1989) is usually cited to illustrate this approach, whereas Chow, Zeigler, and Kim (1994) have applied it in the PDEVs formalism.

2.3. Parallel Replications

Several replications of a sequential simulation are executed in different processors.

Recent work on this topic includes (Biles and Kleijnen 2005).

2.4. Simulation Cloning

Cloning is an emerging approach which “enables the evaluation of multiple simulated futures” (Hybinette and Fujimoto 2001). A running simulation is dynamically cloned at so-called “decision points”, thus

allowing the parallel exploration of different execution paths.

2.5. Domain Decomposition

Domain Decomposition is based on the idea that any simulation model can be decomposed either spatially or temporally.

Time parallel simulation decomposes the time axis and assigns each time interval to a different processor. Afterwards, the results from each simulation interval are combined into an overall solution.

Spatial parallelism sees the physical system as a set of inter-communicating physical processes (PP), and it is considered the classical approach to PDES, as stated in (Fujimoto 2000). When modelling by using this approach, each PP is mapped to a Logical Process (LP). From an event perspective, each LP handles its own (internal) event list and interacts with other LPs by exchanging (external) time-stamped events. In this scenario, event causality makes synchronization among LPs a challenging problem. Two branches of algorithms deal with synchronizing events among LPs: optimistic, such as Time-warp (Jefferson 1985), and conservative, such as Chandy-Misra (Chandy and Misra 1978; Chandy and Misra 1981). An up-to-date survey on this topic can be found in (Perumalla 2006).

An interesting mixture of this approach and Parallel Replication can be found in (Bononi, Bracuto, D'Angelo, and Donatiello 2005).

3. JAVA: IS IT GOLD ALL THAT GLITTERS?

Almost any paper presenting a tool developed in Java, strongly emphasizes the many advantages of this language, often leaving out some not so favorable issues.

Actually, Java has many criticisms. Finding such criticisms out of informal discussions or blogs is a difficult task: on the one hand, people who want to use Java only try to highlight the advantages of the language and minimize its disadvantages in a more or less subtle fashion; on the other hand, people who do not want to use Java simply do not care about it!

Really critical papers can be found only by going back to the 90s. A good example of an excellent critical paper is (Lewis 1997), who tried to slow down Java's exaggerated expectations from a set of smart assertions. Most of these assertions, completely accurate when they were stated, are nowadays, though, superseded.

The rest of this section presents some of the most important Java features trying to highlight both strengths and weaknesses.

3.1. Object Orientation

Object orientation (OO) has been traditionally brandished as a very suitable paradigm for supporting Discrete Event Simulation (Pidd 1995). Indeed, the first object-oriented language, SIMULA (Dahl and Nygaard 1966), was developed to create and run simulation models. Furthermore, Fujimoto (1993) insists on the similarities between PDES and OO.

OO is usually stated as a very easy-to-use paradigm, prone to modularity and reusability. Certainly, Java as an OO language offers these facilities but only if coherently used. Java opponents would argue that a simple program becomes a terrific amount of classes... and maybe they would be right. OO fundamentalists would suffer from class overhead, which can evolve into poor performance.

3.2. Portability

“Write once, run everywhere”: Sun Microsystems' slogan is maybe Java's winning hand. Almost any system has a corresponding Java Virtual Machine (JVM): Linux, Windows, Solaris, Mac OS, HP-UX, cell phones, routers, washing machines... Portability means that a developer can focus on high level characterization and forget hardware constraints such as word size, register file size, and operating system.

Eventually, this slogan has evolved into a kind of a joke: “write once, DEBUG everywhere”. Certainly, things are far from ideal. Even using standard and core Java classes, results are sometimes irreproducible from one machine to another due to subtle differences in the JVM implementations. A good example is the use of threads, whose execution order heavily relies on the Operating System scheduler. Developers interested in “reproducibility” of simulations must be specially careful.

3.3. Lack of Pointers: the Garbage Collector

Java avoids explicit memory management from the user. Thus, memory leaks and bad address arithmetic completely disappear, and Java stands as a safer language that, for example, C or C++. Not only this, the lack of pointers improves the security of the language. The automatic mechanism used for handling memory, the garbage collector (GC), allows the compiler and JVM for further optimization.

The counterpart is that GC makes execution time of the programs highly unpredictable. The frequency of invocation and the time it takes to manage the memory may change drastically from one execution to another.

Due to this lack of user control, Java is also a memory glutton. “Java Heap Space Out of Memory” errors are a nightmare for simulation developers as soon as the model slightly grows.

Readers interested in a deeper insight on memory overhead and the performance impact of the GC may review (Blackburn, Cheng, and McKinley 2004; Kim and Hsu 2000).

3.4. Multi Threading

Java threads are directly built into the language. Actually, Java core support for threads can be considered as simple and elegant, but also quite rudimentary. However, Java 1.5 included an extended package, `java.util.concurrent`, with thread pools, a framework for asynchronous task execution, “collection” type classes optimized for concurrent access, synchronization utilities such as semaphores,

atomic variables, blocks and condition variables. Such an amount of features brings the user the opportunity to exploit multithreading programming in an easy fashion.

Threads can be a powerful and flexible tool if sensibly used. There is a threatening temptation in trying to overexploit multithreading in Java process-oriented simulators. As Jacobs and Verbraeck (2004) states, threads are expensive with regards to computer resources and, instead of solving problems, they may provide enough rope to hang the programmer.

3.5. Network Aware

Java was created with network communications in mind. Several features can be highlighted that make Java network aware:

- Java supports network communication via sockets and provides support for higher level protocols, such as HTML and FTP. The basic socket interface is portable but not efficient enough to be effectively ported to other protocols different from TCP/IP.
- Remote Method Invocation (RMI) is an extra abstraction layer over sockets. RMI enables a method of a class located in a remote computer to be invoked almost transparently from another computer. RMI is sometimes accused of inefficient and unnecessarily complex. Thus, Philippsen, Haumacher, and Nester (2000) presented an alternative design called KARMI, which also includes an alternate serialization mechanism.
- Java objects can be “serialized”, that is, disassembled, sent through a communication stream, and finally assembled again. However, object serialization has several limitations. Thus, objects tied to the Operating System, such as threads, can not be serialized.
- Java Applets are little applications which can be run in a web browser. In spite of their limited access to the user’s system and their significant start-up time, Applets are still a widespread option to develop client front-ends.
- A more up-to-date technology is the Java Web Start (or javaWS) framework. As opposite to Applets, Web Start applications do not run inside the browser. Thus, they overcome most of Applet’s limitations. Moreover, the download time has been reduced by using a compression system.

3.6. Java Lacks

There are some language-specific shortages which a programmer would want to get improved:

- Support for multidimensional arrays. In spite of some efforts in this direction (Moreira, Midkiff, and Gupta 2003), there is no standard Java mechanism for implementing multidimensional arrays, which are an

inestimable tool for optimization for numerical programmers.

- Support for multiple inheritance. This is a controversial issue since many developers claim that the “interface” mechanism makes the trick, whereas other ones insist on the necessity of a more straightforward implementation. Different approaches which try to deal with this problem and further discussion can be found in (Tempero and Biddle 2000; Bettini, Loreti, and Venneri 2002).
- Support for operator overloading. The lack of this feature precludes a user from creating arithmetic classes as efficient as the built-in ones.
- Java Native Interface (JNI) overhead. In spite of being a very powerful tool, JNI introduces an undesirable overhead which may penalize performance (Kurzyniec and Sunderam 2001)
- Benchmarking. When developing software with Java and looking for performance, execution time is generally the only tool for measuring the impact of a so-called improvement. Other general purpose languages such as C or C++, lacking of a virtual machine allows the user for a more strict control of the execution environment. On the contrary, Java introduces extra abstraction layers which contribute extra factors for time measuring. Blackburn, Guyer, Hirzel, et al. (2006) propose a complete methodology for supporting Java software benchmark.

Language extensions such as SPAR (van Rееuwijk, Kuijlmаn, and Sips 2003) have been proposed that deal with some of these limitations.

4. JAVA PERFORMANCE: THE GREAT URBAN LEGEND

Performance has been postulated as the Java’s greatest contender from its very beginning. Nowadays, any misconception about performance should be out of doubt but, unfortunately, urban legends are hard to exile.

The source of such a misconception comes from the interpreted nature of the language. Portability has a price: Java is not directly executed on a computer but on a JVM. A virtual machine is an extra abstraction layer, therefore execution should be penalized.

The appearance of the Just in Time (JIT) compiler changed this situation. Modern JVMs, such as Sun’s HotSpot, carries out runtime analysis to identify critical sections of code. These sections are compiled to native code and adaptively optimized.

Does it mean that Java is as fast as a purely compiled language, such as C or C++? Recently, many benchmarks have appeared which try to answer this question (Bull, Smith, Ball, et al. 2003; Shafi, Carpenter, Baker, and Hussain 2009). The results

clearly show that differences in execution time are almost negligible. Moreover, Java achieves better performance for some tests. In conclusion, performance should not be used as a critical reason for choosing one or another language.

Nonetheless, startup time stills a major drawback for Java when compared to C or C++. This penalty can only be avoided in subsequent executions of a program on the same JVM, when JIT have acquired enough information for performing optimizations. Most benchmarks use these subsequent executions to compare Java and C performance, which could be misleading.

5. JAVA FOR DISTRIBUTED PROGRAMMING

Early works on Java for heterogeneous parallel computing (Dietz 1998; Weems 1998) promoted Java as an emerging option for such systems. Portability, thread- and network- aware design, object orientation... were highlighted as some of the reasons for using Java. Nowadays, the proliferation of tools, frameworks and libraries has corroborated this initial mind, and it seems that any distributed environment may profit from this language. Thus, all of the PDES approaches, from parallel replication to domain decomposition, could be built on top of these Java parallel frameworks and libraries.

Factor, Schuster, and Shagin (2004) make a comprehensive survey on this topic, but there are many other recent tools which are not included in their paper such as MPJ (Carpenter, Getov, Judd, et al. 2000), JavaNOW (Thiruvathukal, Dickens, and Bhatti 2000), CCJ (Nelisse, Maassen, Kielmann, and Bal 2003), Ibis (van Nieuwpoort, Maassen, Wrzesinska, et al. 2005), JOPI (Al-Jaroodi, Mohamed, Jiang, and Swanson 2005), JavaSymphony (Fahringer and Jugravu 2005), MPJE, Tycho (Baker, Grove, and Shafi 2006), JCluster (Zhang, Yang, and Zheng 2006), Parallel Java (Kaminsky 2007), JaMP (Klemm, Bezold, Veldema, and Philippsen 2007), Hydra (Powers and Alaghband 2008), and Babylon (van Heiningen, Macdonald, and Brecht 2008).

6. JAVA FOR PARALLEL DISCRETE EVENT SIMULATION

A literature survey returns a prolific Java-based software production for DES. A few examples, which can be currently downloaded and used, are:

- JSIM (Miller, Nair, Zhang, and Zhao 1997)
- simJava (Howell and Mcnab 1998)
- Simkit (Buss 2002)
- javaSimulation (Helsgaun 2004)
- PsimJ (Garrido and Im 2004)
- SSJ (L'ecuyer and Buist 2005)
- DESMO-J (Page and Kreutzer 2005)
- JAPROSIM (Abdelhabib and Brahim 2008)
- SIGHOS (Castilla, García, and Aguilar 2009)

Java-based PDES are not as numerous, but there are many interesting examples, which will be overviewed in the rest of this section.

6.1. JUST: Java Ubiquitous Simulation Tools

Cassel and Pidd (2001) propose a distributed DES based on the Three-Phase approach (Pidd 1995).

The initial version of the tool was a client-server package: the server managed the simulation and the clients have the implementation of the model.

A second version, JUSTDistributed, is a PDES with conservative synchronization, featuring null message and lookahead. JUSTDistributed is multithreaded and uses RMI for communication. The authors deal with the resource contention and shared states problems, and propose a solution based on a central repository which reduces the message load.

6.2. SPADES/Java

SPaDES /Java (Teo and Ng 2002) is a PDES library, intended to isolate the user from the implementation details regarding event synchronization and parallelization.

This process-oriented simulator supports both sequential and parallel execution. The later uses a conservative null-message protocol for synchronization.

When modelling with the library, the conceptual model consists of permanent and temporary processes. Resources are treated as permanent processes and modelled as LPs; whereas any other entity is treated as a temporary process and modelled as a time-stamped message passed between LPs. Actually, only permanent processes are implemented as Java threads.

Programming a parallel simulator with SPaDES/Java requires using a template comprising four main parts: definitions of the processes, process routines that define the simulation logic, code for initializing and starting the simulator, and message abstraction and reconstruction routine for the case of parallel simulation.

6.3. D-SOL

Jacobs, Lang, and Verbraeck (2002) presents a fully distributed simulation environment called D-SOL. This tool is an example of a dedicated execution simulator.

The simulation framework consists of two remote network services: an event-based DES remote simulation service; and a system representation service, which includes representation and statistical libraries. Such service-oriented architecture is intended to enhance flexibility, distribution of simulation models, and integration and interaction with real-time information systems.

The simulation library core is event-based, since the authors insist on the advantages of such world-view over the activity-scanning and the process-interaction ones (Balci 1988). However, a porting of the process interaction approach to be supported by the event-based core is presented (Jacobs and Verbraeck 2004). The "stack swapping" technique is used to obtain a single-

threaded implementation of the process interaction approach. Such implementation relies on a Java interpreter built on top of the JVM, which is used any time a “pausable method”, that is, a process that requires to be stopped now and resumed later, is invoked. The interpreter makes process-based models around 6 times slower than pure event-based ones, when there are more than 1000 simultaneously created entities. Time penalty is even higher when creating fewer entities.

Last versions of D-SOL, as explained in (Jacobs 2005), include multiformalism support. Thus, the simulation framework allows a user to base its development in any of DEVS, DESS, and mixed DEVS-DESS formalisms (Zeigler, Praehofer, and Kim 2000).

6.4. CSA&S/PV: Complex Systems Analysis & Simulation – Parallel Version

(Niewiadomska-Szynkiewicz, Zmuda, and Malinowski 2003) have developed a parallel software environment based on an earlier design programmed in C. This tool is intended to aid a user when designing and simulating complex physical processes.

The parallelism is exploited by using a conservative LP approach, but the user can configure each LP local clock to be event-driven, or time-driven (i.e., a clock which advances at regular intervals).

The developed tool consists of five modules:

- Shell. The user graphical interface.
- Calculation module or manager: The system kernel, which handles computation and inter-process communication.
- Communication library: An interface that provides communication facilities between the shell and the calculation module.
- User application: The simulation model as defined by the user. The LPs are defined in this module.
- User library: A set of methods to communicate the user application and the manager.

CSA&S/PV has been mainly applied to computer and water networks, but it is intended to be a general purpose simulation tool.

6.5. Summary

Table 1 summarizes the characteristics of the tools introduced formerly in this section, plus further parallel simulators: JTED (Cowie 1998), Fornax (Halderen and Overeinder 1998), and JWarp (Bizarro, Silva, and Silva 1998).

Most of these simulators exploit spatial domain decomposition, that is, the classical LP approach. Only D-SOL follows a different path and focuses on dedicated execution. The most (negatively) remarkable issue is that only two of the tools are currently available to be downloaded and used.

7. WHAT IS NEXT?

Java has a very active community and the language is continuously evolving. New enhanced features and language improvements are coming.

- Modularity. Java’s standard library is often accused of being not only big, but huge. In spite of the advantages regarding compatibility and portability, many programmers find simply impossible to exploit coherently such an amount of core classes. Modularity seems to be the answer. Thus, application would be installed with just those components actually required from the Java standard library. Not only this, the new proposal seems to add a better handling mechanism for dependence and visibility of packages and classes.
- File access. An improved file-access API is planned which will overcome most of the shortages of the current one, such as support for symbolic links, atomic move and copy operations, improved permission control...
- Improved date and time support. Representations of date without time, time without date, durations and intervals will be included. Simulator developers will see immediately the potential benefits of this feature in terms of portability, interconnection of different simulation components, and easiness of design.
- Units and quantities. Several packages would be added for managing physical quantities and their expression as numbers of units. Once again, profits for simulation development are immediate.
- More parallel processing. A new parallel processing package with fork/join primitives will be included which allows the programmer to work in divide and conquer style algorithms. Moreover, better tools for handle parallel arrays will be added. PDES could exploit these new features.
- Graphical Interface. The Swing Application framework (SAF) is intended to simplify building desktop applications.
- G1 GC. A brand new Garbage Collector, with enhanced performance is being designed.

Other proposals include a language level support for XML, which would improve the XML-Java integration. The official source for the current state of development is the JDK 7 Project website (<https://jdk7.dev.java.net/>). Several blogs and forums have “unofficial” information about what is expected to be included in further Java versions: Alex Miller’s blog (<http://tech.puredanger.com/java7/>) contains a very comprehensive list of proposed features; JavaWorld publishes thousands of Java articles with up-to-dated information, such as (Miller 2008).

Table 1: Summary of PDES Tools

PDES Tool	Parallelism exploitation	World-view	Available
JUST	Domain Decomposition (Conservative)	Three-Phase	-
SPaDES/Java	Domain Decomposition (Conservative)	Process-oriented	http://www.comp.nus.edu.sg/~pasta/spades-java/spadesJava.html
D-SOL	Dedicated Execution	Multiformalism	http://sk-3.tbm.tudelft.nl/simulation/index.php
CSA&S/PV	Domain Decomposition (Conservative)	Event-based	-
JTED	Domain Decomposition (Conservative)	Event-based	-
Fornax	Domain Decomposition (Optimistic and Conservative)	Process-oriented	-
JWarp	Domain Decomposition (Optimistic)	Event-based	-

8. CONCLUSIONS

Undoubtedly, Java has reached its maturity and it is a convenient programming language to be used across a wide spectrum of applications.

Even the performance shortfall, which has been traditionally argued as a major Java drawback with regards to other programming languages such as C and C++, is losing weight. Recent benchmarks have shown that this assertion is far from the truth. In short, Java outperforms or underperforms C and C++ depending on the specific application.

Numerous examples have shown the suitability of using Java for Discrete Event Simulation and Parallel Discrete Event Simulation. To the best of our knowledge, only two Java-based parallel simulators are currently available to be freely downloaded. Being not part of this survey, we have observed that non-Java based PDES show a similar behaviour, and not many are freely (neither commercially) available.

Java seems to be a good option though it still poses several shortages and there is a long road ahead for improvements. Of course, C and C++ have not said their last word. Moreover, there is still a gap in simulation for Microsoft .NET, dynamic languages (Ruby), scripting languages such as Python and Perl, and many other state-of-the-art programming languages. Only time will tell if they will succeed or not.

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

Abdelhabib, B. and Brahim, B., 2008. JAPROSIM: A Java framework for Process Interaction Discrete Event Simulation, *Journal of Object Technology*, 7(1), 103-119.

Al-Jaroodi, J., Mohamed, N., Jiang, H., and Swanson, D., 2005. JOPI: a Java object-passing interface,

Concurrency and Computation: Practice and Experience, 17(7-8), 775-795.

- Baker, M., Grove, M., and Shafi, A., 2006. Parallel and Distributed Computing with Java, *Fifth International Symposium on Parallel and Distributed Computing*. IEEE, 3-10.
- Balci, O., 1988. The implementation of four conceptual frameworks for simulation modeling in high-level languages, *Proceedings of the 20th conference on Winter simulation - WSC '88*, New York, ACM Press, 287-295.
- Bettini, L., Loreti, M., and Venneri, B., 2002. On Multiple Inheritance in Java, *Proc. of TOOLS EASTERN EUROPE, Emerging Technologies, Emerging Markets*, 1-15.
- Biles, W. E. and Kleijnen, J. P., 2005. International Collaborations in Web-based Simulation: A Focus on Experimental Design and Optimization, *Proceedings of the Winter Simulation Conference*, IEEE, 218-222.
- Bizarro, P., Silva, L. M., and Silva, J. G., 1998. JWarp: a Java library for parallel discrete-event simulations, *Concurrency: Practice and Experience*, 10(11-13), 999-1005.
- Blackburn, S. M., Cheng, P., and McKinley, K. S., 2004. Myths and realities: The Performance Impact of Garbage Collection, *ACM SIGMETRICS Performance Evaluation Review*, 32(1), 25-36.
- Blackburn, S. M., Guyer, S. Z., Hirzel, M., Hosking, A. L., Jump, M., Lee, H., et al., 2006. The DaCapo benchmarks: java benchmarking development and analysis, *Proceedings of the 2006 OOPSLA Conference*, ACM, 169.
- Bononi, L., Bracuto, M., D'Angelo, G., and Donatiello, L., 2005. Concurrent Replication of Parallel and Distributed Simulations, *Workshop on Principles of Advanced and Distributed Simulation (PADS'05)*, 234-243.
- Bull, J.M., Smith, L.A., Ball, C., Pottage, L., and Freeman, R., 2003. Benchmarking Java against C and Fortran for scientific applications, *Concurrency and Computation: Practice and Experience*, 15, 417-430.

- Buss, A., 2002. Component based simulation modeling with simkit, *Proceedings of the Winter Simulation Conference*, 243-249.
- Carpenter, B., Getov, V., Judd, G., Skjellum, A., and Fox, G. C., 2000. MPJ: MPI-like message passing for Java, *Concurrency: Practice and Experience*, 12(11), 1019-1038.
- Cassel, R. A. and Pidd, M., 2001. Distributed discrete event simulation using the three-phase approach and Java, *Simulation*, 8, 491-507.
- Castilla, I., García, F.C., and Aguilar, R.M., 2009. Exploiting concurrency in the implementation of a discrete event simulator, *Simulation Modelling Practice and Theory*. 17(5), 850-870, doi:10.1016/j.simpat.2009.02.006.
- Chandy, K.M., and Misra, J., 1978. Distributed simulation: a case study in design and verification of distributed programs, *IEEE Transactions on Software Engineering*, 5(5), 440-452.
- Chandy, K.M., and Misra, J., 1981. Asynchronous distributed simulation via a sequence of parallel computations, *Communications of the ACM*. 24(4), 198-205.
- Chow, A. C., Zeigler, B. P., and Kim, D. H., 1994. Abstract simulator for the parallel DEVS formalism, *Fifth Annual Conference on AI, and Planning in High Autonomy Systems*. Gainesville, FL, USA, IEEE Comput. Soc. Press, 157-163.
- Comfort, J.C., 1984. The simulation of a master-slave event set processor, *Simulation*, 42(3):117-124.
- Concepcion, A.I., 1989. A hierarchical computer architecture for distributed simulation, *IEEE Transactions on Computing*, 38(2), 311-319.
- Cowie, J., 1998. Jted: parallel discrete-Event simulation in java, *Concurrency: Practice and Experience*, 10(11-13), 993-997.
- Dahl, O., and Nygaard, K., 1966. SIMULA: an ALGOL-based simulation language. *Communications of the ACM*. 9(9), 671-678
- Dietz, H., 1998. Heterogeneous parallel computing with Java: jabber or justified?, *Proceedings Seventh Heterogeneous Computing Workshop (HCW'98)*, IEEE Comput. Soc, 159-162.
- Factor, M., Schuster, A., and Shagin, K., 2004. A distributed runtime for java:yesterday and today, *Proceedings of the 18th International Parallel and Distributed Processing Symposium.*, IEEE, 159-165.
- Fahringer, T. and Jugravu, A., 2005. JavaSymphony: a new programming paradigm to control and synchronize locality, parallelism and load balancing for parallel and distributed computing, *Concurrency and Computation: Practice and Experience*, 17(7-8), 1005-1025.
- Fujimoto, R.M., 1993. Parallel discrete event simulation: will the field survive? *ORSA Journal on Computing*. 5(3), 213-230.
- Fujimoto, R.M., 2000. *Parallel and Distributed Simulation Systems*, Wiley Interscience.
- Garrido, J.M., and Im, K., 2004. Teaching Object-Oriented Simulation with PsimJ Simulation Package, *Proceedings of the 42 Annual ACM Southeast Regional Conference*, 422-427, Huntsville, AL.
- Goes, L., Pousa, C., Carvalho, M., Ramos, L., and Martins, C., 2005. JSDESLib: A Library for the Development of Discrete-Event Simulation Tools of Parallel Systems, *19th IEEE International Parallel and Distributed Processing Symposium*. IEEE, 184b-184b.
- Halderen, B. A. and Overeinder, B. J., 1998. Fornax: Web-based distributed discrete event simulation in Java, *Concurrency: Practice and Experience*, 10(11-13), 957-970.
- Helsgaun, K., 2004. Discrete Event Simulation in Java, Technical Report, Roskilde University.
- Howell, F. and Mcnab, R., 1998. simjava: a discrete event simulation library for java, *International Conference on Web-Based Modeling and Simulation.*, 51-56.
- Hybinette, M., and Fujimoto, R.M., 2001. Cloning parallel simulations, *ACM Transactions on Modeling and Computer Simulation*, 11(4), 378-407.
- Jacobs, P.H.M., 2005. The DSOL simulation suite - Enabling multi-formalism simulation in a distributed context, Dissertation, TU Delft.
- Jacobs, P.H.M., Lang, N., and Verbraeck, A., 2002. D-Sol; a distributed java based discrete event simulation architecture, *Proceedings of the Winter Simulation Conference*. IEEE, 793-800.
- Jacobs, P.H.M., and Verbraeck, A., 2004. Single-Threaded specification of process-Interaction formalism in java, *Proceedings of the 2004 Winter Simulation Conference*, IEEE, 479-486.
- Jefferson, D., 1985. Virtual time, *ACM Transactions on Programming Languages and Systems*. 7(3), 404-425.
- Kaminsky, A., 2007. Parallel Java: A Unified API for Shared Memory and Cluster Parallel Programming in 100% Java, *2007 IEEE International Parallel and Distributed Processing Symposium*. IEEE, 1-8.
- Kim, J., and Hsu, Y., 2000. Memory system behavior of Java programs, *Proceedings of the 2000 ACM SIGMETRICS international conference on Measurement and modeling of computer systems - SIGMETRICS '00*, New York: ACM Press, 264-274.
- Klemm, M., Bezold, M., Veldema, R., and Philippsen, M., 2007. JaMP: an implementation of OpenMP for a Java DSM, *Concurrency and Computation: Practice and Experience*, 19(18), 2333-2352.
- Kurzyniec, D., and Sunderam, V., 2001. Efficient cooperation between Java and native codes - JNI performance benchmark, *The 2001 International Conference on Parallel and Distributed Processing Techniques and Applications*, Las Vegas, Nevada.

- L'ecuyer, P. and Buist, E., 2005. Simulation in java with ssj, *Proceedings of the Winter Simulation Conference*. IEEE, 611-620.
- Lewis, T., 1997. If Java is the answer, what was the question? *Computer*, 30, 133-135.
- Miller, A., 2008. *Year in Review: What to expect in Java SE 7*, JavaWorld. Available from: <http://www.javaworld.com/javaworld/jw-12-2008/jw-12-year-in-review-2.html> [accessed 23 April 2009].
- Miller, J. A., Nair, R., Zhang, Z., and Zhao, H., 1997. JSIM: A Java-based simulation and animation environment, *Proceedings of 1997 SCS Simulation Multiconference*, 31-42.
- Moreira, J.E., Midkiff, S.P. and Gupta, M., 2003. Supporting multidimensional arrays in Java, *Concurrency and Computation: Practice and Experience*, 15, 317-340.
- Nelisse, A., Maassen, J., Kielmann, T., and Bal, H. E., 2003. CCJ: object-based message passing and collective communication in Java, *Concurrency and Computation: Practice and Experience*, 15(35), 341-369.
- Niewiadomska-Szynkiewicz, E., Zmuda, M., and Malinowski, K., 2003. Application of a java-based framework to parallel simulation of large-scale systems, *Applied Mathematics and Computation*, 13(4), 537-547.
- Page, B., and Kreutzer W., 2005. *The Java Simulation Handbook: Simulating Discrete Event Systems with UML and Java*, Shaker Publishing.
- Page, E. H., Moose, R. L., and Griffin, S. P., 1997. Web-based Simulation In Simjava Using Remote Method Invocation, *Proceedings of the 1997 Winter Simulation Conference*. IEEE, 468-474.
- Perumalla, K.S., 2006. Parallel and distributed simulation: traditional techniques and recent advances, *Proceedings of the 38th Conference on Winter Simulation*, Monterey, California, 84-95.
- Philippsen, M., Haumacher, B., and Nester, C., 2000. More efficient serialization and RMI for Java, *Concurrency: Practice and Experience*, 12(7), 495-518.
- Pidd, M., 1995. Object orientation, discrete simulation and the three-phase approach, *Journal of the Operational Research Society*, 46, 362-374
- Powers, F. E. and Alaghband, G., 2008. The Hydra Parallel Programming System, *Concurrency and Computation: Practice and Experience*, 20(1), 1-27.
- Shafi, A., Carpenter, B., Baker, M., and Hussain, A., 2009. A comparative study of Java and C performance in two large-scale parallel applications, *Concurrency and Computation: Practice and Experience*, in press. doi:10.1002/cpe.1416.
- Tempero, E. and Biddle, R., 2000. Simulating multiple inheritance in Java, *Journal of Systems and Software*, 55, 87-100.
- Teo, Y. M. and Ng, Y. K., 2002. Spades/java: object-Oriented parallel discrete-Event simulation, *Proceedings 35th Annual Simulation Symposium*, 245-252.
- Thiruvathukal, G. K., Dickens, P. M., and Bhatti, S., 2000. Java on networks of workstations (JavaNOW): a parallel computing framework inspired by Linda and the Message Passing Interface (MPI), *Concurrency: Practice and Experience*, 12(11), 1093-1116.
- van Heiningen, W., Macdonald, S., and Brecht, T., 2008. Babylon: middleware for distributed, parallel, and mobile Java applications, *Concurrency and Computation: Practice and Experience*, 20(10), 1195-1224.
- van Nieuwpoort, R. V., Maassen, J., Wrzesinska, G., Hofman, R. F., Jacobs, C. J., Kielmann, T., Bal, H.E., 2005. Ibis: a flexible and efficient Java-based Grid programming environment, *Concurrency and Computation: Practice and Experience*, 17(7-8), 1079-1107.
- van Reeuwijk, C., Kuijman, F. and Sips, H.J., 2003. Spar: a set of extensions to Java for scientific computation, *Concurrency and Computation: Practice and Experience*, 15, 277-297.
- Weems, C.C., 1998. Heterogeneous programming with Java: gourmet blend or just a hill of beans?, *Proceedings Seventh Heterogeneous Computing Workshop (HCW'98)*, IEEE Comput. Soc, 173-182.
- Zeigler, B., Praehofer, H., and Kim, T., 2000. *Theory of modeling and simulation*. Academic Press, San Diego: CA, USA, 2nd edition.
- Zhang, B., Yang, G., and Zheng, W., 2006. Jcluster: an efficient Java parallel environment on a large-scale heterogeneous cluster, *Concurrency and Computation: Practice and Experience*, 18(12), 1541-1557.

AUTHORS BIOGRAPHY

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 research interests include parallel discrete event simulation and computer architecture.

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. Her current research interests are decision making based on discrete event simulation systems and knowledge-based systems, intelligent agents, and intelligent tutorial systems.

MODBUS BASED PROTOCOL FOR COMMUNICATION BETWEEN A DISTRIBUTED CO-SIMULATION BACKBONE AND REAL ELEMENTS FOR SIMULATION MATTERS

Tales Marchesan Chaves⁽¹⁾, Bráulio Adriano de Mello⁽²⁾, Paulo Betencourt⁽¹⁾

⁽¹⁾URI – DECC – Santo Ângelo – RS - Brazil

⁽²⁾UFLA – Universidade Federal de Lavras – Lavras – MG - Brazil

⁽¹⁾tales.chaves@gmail.com, ⁽²⁾bmello@dcc.ufla.br, ⁽¹⁾pbetencourt@urisan.tche.br

ABSTRACT

Heterogeneous simulation allows for the combination of real and virtual elements for simulation matters. The DCB (Distributed Cosimulation Backbone) is a cosimulation software that supports simulation of heterogeneous models. Integration of real and virtual elements is advantageous when some parts of a system are hard to simulate. However, the adaptation of real elements into the DCB is hard and time consuming. This paper presents a protocol used for communication between the DCB and real elements. This paper also presents a framework that facilitates the adaptation process.

Keywords: heterogeneous simulation, real elements, protocol, framework.

INTRODUCTION

Heterogeneous simulation consists of the execution of a model comprising different elements, described in different levels of abstraction, whose main purpose is to imitate a real world process or operation (Reynolds 1988). Heterogeneous models can combine simulated or real elements in order to perform a expected function determined by the simulation model. These models are appropriated when experiments are not possible or imply higher costs.

The number of attributes that each element has makes its structure more complex. This reflects into the simulation model resulting in models that are harder to be modeled, as exemplified in (Rafull, Queiroz, Souza, and Pinto 2006). This example demonstrates the creation of a simulation model of an automatic control system for harvesting machine cutting height. The construction of this model requires the use of differential equations which were used to analyze a number of factors such as: the speed of the machine; hydraulic-system supply pressure; the controller reference contact force.

Systems like these increase the cost and difficulty of the modeling process. To construct such a model, experience and expertise are required what creates the necessity of hiring more expensive designers.

The amount of elements being simulated to represent such a system increases and in the process elevates memory and processing use.

By adopting heterogeneous simulation as an alternative to model systems, both issues mentioned can be addressed by using real elements instead of modeling them as virtual elements. The use of real elements, in this case, (in place of virtual elements usually created from scratch) reduces the necessary effort needed to construct part of the model. Also, as part of the model gets executed in real elements, the processing load of the simulation model, decreases as a consequence of less use of virtual elements.

Notwithstanding the advantages of using real elements in the modeling of a system, some problems may arise. For instance, the bandwidth of the bus used for data transmission might not be enough for data traffic between the real element and the simulator. If a real element generates constant data at a faster speed than the bus can handle, important data might be lost, compromising simulation results. This problem can be solved by buffer utilization. A better approach to solve it would be the utilization of a communication protocol that implements acknowledge primitives (E.g. ACKs). These primitives are used to indicate the successful receipt of a packet.

Another problem to be considered is the difficulty in detecting errors in real elements. The hardware that composes real elements might not be working properly. Again, use of a communication protocol can address the issue by using *timeouts*. In other words, one can conclude a real element is not working if it does not respond for a reasonable length of time. (Heiko 1998) describes other problems likely to occur when using real elements.

Besides the problems of integrating real elements, heterogeneous models require support architecture capable of implementing cooperation between different elements. The paradigm of heterogeneous models must be supported by the simulator software. The DCB (Distributed Co-simulation Backbone) (Mello, Souza, Sperb, and Wagner 2005) architecture supports models that integrate heterogeneous elements. The solutions developed by this work target the DCB.

Nevertheless, the adaptation of real elements into the DCB is awkward and requires from the designer a profound knowledge of composition policies and simulation on the DCB software.

This work attempts to facilitate the integration of real elements into the DCB. To accomplish this goal, this article presents a communication protocol, based on the MODBUS standard, that formalizes data transfer between DCB and real elements. This protocol addresses some issues related to data transfer between real elements and DCB, such as data loss; device addressing and hardware malfunction detection. These features are essential to maintain simulation accuracy.

This paper is organized as follows. Section 2 presents a description of adaptation process. Section 3 provides an overview of the MODBUS standard. Section 4 presents the solution developed for this work. Section 5 describes a case study in which a rice-seed counter sensor was adapted into the DCB and Section 6 presents the conclusions of the paper.

ELEMENT ADAPTATION INTO THE DCB

According to (Fummi, Loghi, Poncino, and Pravadelli 2009) there is no existing methodology for heterogeneous co-simulation environments. According to (Oraw, Choudhary, and Ayyanar 2007) , in the future, the biggest challenge to the simulation field will be the capacity of handling model complexity. In order to lessen this complexity, real elements can be utilized. The adaptation of real elements into the DCB is possible due to its architectural support for heterogeneity. A module, member of DCB architecture, called *gateway* (shown in Figure 1 as a component of the DCB architecture), provides an interface to external elements.

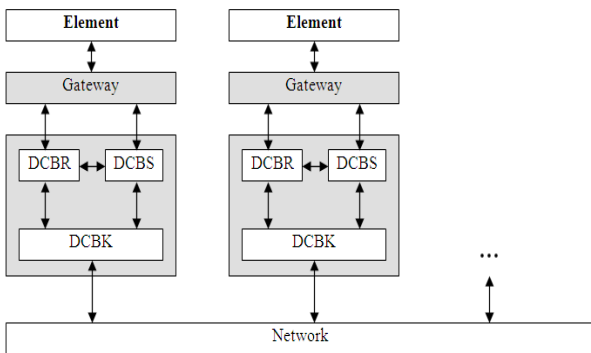


Figure 1. DCB architecture

The *gateway* module provides a structure that allows for each element to include specific information about its behavior (E.g. an algorithm that describes the behavior of the element), inputs and outputs. Although, to include the description of the element, the designer must have an understanding of the inner details of the DCB structure. This fact makes more difficult the adaptation of elements because it exposes some inner details of the DCB implementation. Therefore, before adapting an element, the designer must understand how the DCB accesses and controls external elements.

When adapting a real element to the DCB software, the designers must also implement a driver to handle the interface used by the element. The DCB does not provide any facilities concerning this issue.

Thus, to facilitate the adaptation of elements, an API that provides a set of functions for handling the more popular communication interfaces such as USB, RS-232 and Parallel Port is necessary. This API must include functions to configure parameters of the interface, such as speed and parity bit. In addition, functions to send and receive data, together with interface-specific functions, must be present in the API.

Real elements differ in a variety of aspects such as communication interface, and energy consumption. In some cases, the element may require the development of a controller device, used to adapt the element into a computer interface. This diversity creates the need for the development of a new controller for every new real element adapted into the simulation software. Designing and implementing a generic controller capable of handling several different elements is practically impossible.

The next section provides an overview of the MODBUS Protocol. Section 4 presents the solutions, developed by this work, for adapting real elements into the DCB.

MODBUS PROTOCOL

Modbus is an application layer messaging protocol, positioned at level 7 of the OSI model, that provides client/server communication between devices connected on different types of busses or networks (Modbus 2009).

Modbus packet format is shown in Figure 2. It is a request/reply protocol and offers services specified by function codes.

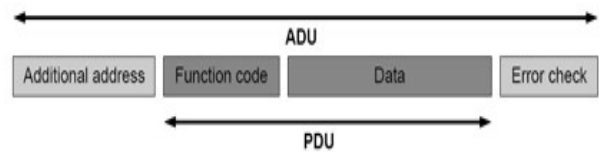


Figure 2. Modbus packet format (Modbus 2009).

SOLUTION FOR ADAPTING REAL ELEMENTS INTO THE DCB

The structure developed to facilitate the adaptation of real elements into the DCB is based on the architecture presented on Figure 3. This architecture is composed of modules involved in the adaptation process such as real element; adaptation hardware; Generic I/O and DCB.

The real element module is composed of the element itself. The Generic I/O module is responsible for handling communication between real elements and the *gateway*. This module is capable of supporting for three communication interfaces: RS-232; Parallel Port; and USB. These interfaces are used to connect the real element to the computer running a DCB instance. Along with the support provided to the interfaces, three

libraries of functions, whose main objective is to control and configure each interface, were developed and included in the Generic I/O module.

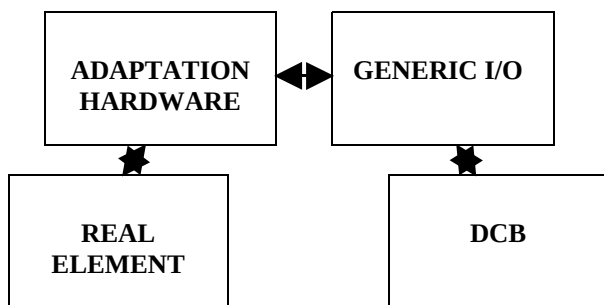


Figure 3. Entities relationship in the process of adaptation of real elements

The adaptation hardware consists of a controller device, described in section 4.1, which was developed to connect a rice-seed counter sensor to the parallel interface. Section 4.2 presents the DCB communication protocol, which is the main focus of this article.

1.1. Real element controller

There is a big number of real elements being used by the industry. They may consist of simple red-light sensors used to detect the presence of an object. Although, some real elements present a complex structure, difficult to be modeled, such as the human body.

This work utilizes mostly real elements composed by sensors or actuators. A controller consists of a special-purpose, customized hardware whose main objective is to provide a communication interface between the DCB and the real element. The real element must be connected to the controller to be able to communicate with the DCB software.

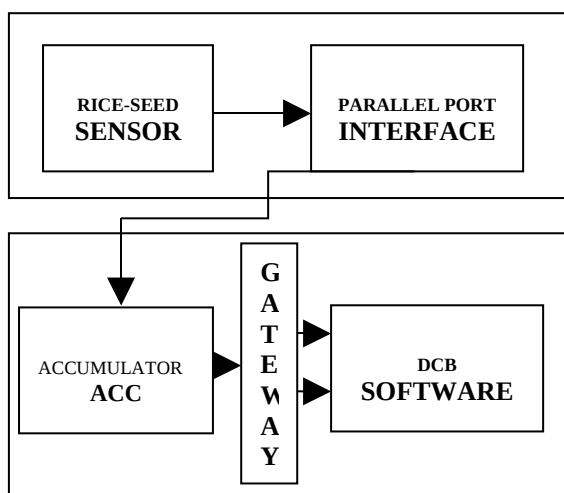


Figure 4. Model-view of the simulated model

The controller constructed in this trial targets a rice-seed counter sensor. It was developed using a

protoboard (prototyping board) which provides a parallel interface for the real element.

The real element is connected to the parallel port through the protoboard (Figure 4). The element consists of an infra-red light sensor that detects a new rice-seed and generates a signal, which is used to increment the counter. The controller, with a real element attached to it, can be used to validate the proposed protocol implementation.

1.2. DCB Communication Protocol

The protocol developed to handle communication between real elements and the DCB is strongly based on MODBUS protocol. The main objective of this protocol is to address typical communication problems that may lead to simulation erroneous results such as data loss; hardware malfunctioning detection; and on-device processing.

The communication protocol structure allows data emission and reception from real elements to the DCB or vice-versa. Also, functions for gathering the current status of elements are defined. The transmitted packet is shown in Figure 2. The supported functions are described in Table 1.

Table 1: DCB Protocol supported functions

Function Code	Function Name	Description
0x04	RIR	Read a single input register
0x06	WIR	Write operation on a single register
0x10	WIRS	Write operation on multiple registers

Whenever an element wants to send data to another element or to the DCB it should first assemble the packet. The 'Additional Address' field indicates the target element. It is important to observe that this feature allows for connection of multiple elements into the DCB. When creating the model, each element has an address assigned to it so that all the packets can be routed to the right destination during the simulation.

The 'Function Code' field contains the function code. The allowed codes are shown in Table 1. The 'Data' field contains the payload. Finally, the 'Error Check' field contains a CRC calculation result which is used to check for packet integrity (Griffiths and Stones 1987).

The following advantages have been found in favor of the adoption of this protocol for handling data transferred during the simulation execution: the possibility of adapting several elements at the same gateway (due to the address field); malfunction check; DCB *overhead* decreasing occurs when real elements exchange request/reply messages without DCB intervention.

CASE STUDY

Precision Agricultural (PA) can benefit from the use of heterogeneous simulation. PA is defined by (McBratney, Bouma, Whelan, and Ancev 2005) as a technique for systematic and organized management of the production system through the information domain. The essence of PA lies in the continuous gathering of information about a particular crop, followed by proper use of this information to optimize maneuvers. Furthermore, PA research is focused on finding new solutions targeting agricultural equipment, thereby justifies the case study developed.

The case study consists of the adaptation of a rice-seed counter sensor into the DCB. The sensor is made up of a set of infra-red lights that detect a rice seed. The number of rice seeds gathered by the equipment is accumulated and transmitted to the DCB at a specific rate. The model being simulated at the DCB software is showed in Figure 5.

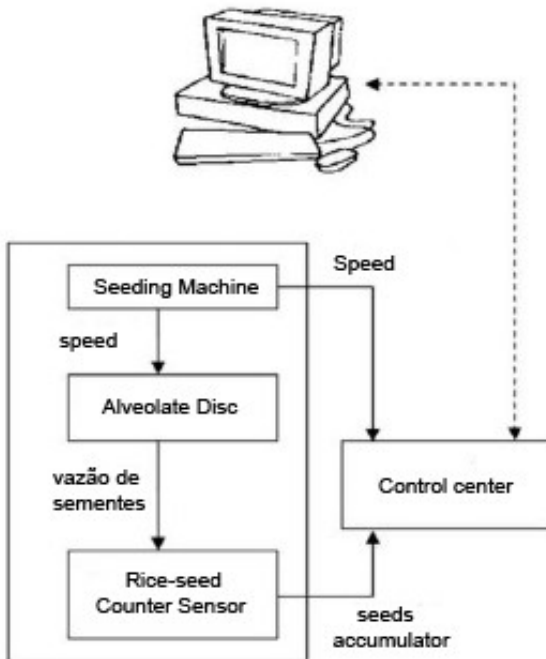


Figure 5. SEP simulation model (Mello and Caimi 2008)

The rice-seed sensor increments the counter whenever an electric pulse arrives at the parallel port. A *driver* was developed to collect incoming data received through the parallel port. This driver was developed in C due to facilities the language provides.

DCB implementation was conceived in Java, giving rise to the need to implement a means of communication between the C *driver* and the DCB running in the JVM (Java Virtual Machine). Process communication was used to circumvent this problem and is exemplified in Figure 6.

```

Runtime runtime = Runtime.getRuntime();
Process proc = null;
try
{
    if (interface_comm == PARALELA)
        proc = runtime.exec("pp read");
    else if (interface_comm == SERIAL)
        proc = runtime.exec("OpenCom read");

    InputStream ip = proc.getInputStream();
    InputStreamReader ir = new InputStreamReader(ip);
    BufferedReader buf_reader = new BufferedReader(ir);

    String lido = null;
    if ((lido == buf_reader.readLine()) != null)
    {
        short x = Short.parseShort(lido);
        return x;
    }
}

```

Figure 6. Process communication in Java

The simulation results of the SEP model are shown in Figure 7.

Tests were carried out to validate the heterogeneous model implemented. According to Figure 7, the input parameters were: speed (km/h); number of holes in the disc; and numbers per revolution. These parameters correspond to numbers 1, 2 and 3 in the Figure 7, respectively. The results (output) of the simulation are: average rate seeds per meter – sensor, speed (meters/sec), instant seeds per meter, corresponding to numbers 4, 5, 6 and 7, respectively.



Figure 7. Simulation results

Here, the simulation adaptation process (using adaptation architecture and DCB communication protocol) is explained as well as its advantages.

6. CONCLUSION

This article presented the issues related to adapting real elements into the DCB software in order to simulate heterogeneous models. Real element adaptation into the

DCB is facilitated by a framework that abstracts the inner details of the DCB software. This framework provides an API of functions operating on communication interfaces. The API provides functions to configure parameters related to a communication interface (such as transmission speed, and data parity), as well as functions for data emission and reception.

The API provided the foundations for the development of a communication protocol, which standardizes communication between real elements and the DCB.

The main features of the protocol are *addressing* field, which permits the adaptation of more than one element per gateway; *error check* field, which enables the element/DCB to verify packet integrity and *function code* field, which describes the function that should be performed on the receiving part.

Therefore, the contributions of this work consist of an organization of the adaptation architecture, which delimits main entities present in models containing real elements. The adaptation architecture helps the designer understand the entities which are part of the process and describes the communication needs of each interface.

Moreover, the DCB communication protocol, which address communication issues, such as data loss, hardware malfunctioning, and stream control, greatly improves simulation accuracy.

The contributions of this work can be applied to several fields, among which, precision agriculture. The development of a more flexible controller, that supports a greater number of real elements remains a topic for future research.

REFERENCES

- Fummi, F., Loghi, M., Poncino, M., and Pravadelli, G. 2009. A cosimulation methodology for hw/sw validation and performance estimation. *ACM Trans. Des. Autom. Elect. Syst.*, vol. 14 Article 23.
- Griffiths, G., Stones, G. C. 1987. The Tea-Leaf Reader Algorithm: An Efficient Implementation of CRC-16 and CRC-32. *Communications of the ACM*. v.30:617-620.
- Heiko, H.: A survey of HW/SW cosimulation techniques and tools. Thesis (M.S.). Royal Institute of Technology, Stockholm.
- McBratney, A., Bouma, J.; Whelan, B., Ancev, T., 2005. Future directions of precision agriculture. *Precision Agriculture* vol.6:1-17.
- Mello, B. A.; Caimi, L. L. 2008. Simulação na validação de sistemas computacionais para a agricultura de precisão. *Rev. bras. eng. agríc. ambient.* vol.12:666-675.
- Mello, B. A.; Souza, U. R. F.; Sperb, J. K.; Wagner, F. R. 2005. Tangram-virtual integration of heterogeneous IP components in a distributed co-simulation environment. *IEEE – Design and Test of Computers*, vol.22:462-471.
- Modbus. 2009. Modbus application protocol specification v1.1b. Modbus Organization.

Available from: <http://www.modbus.org> [accessed 24 May 2009]

- Oraw, B.; Choudhary, V.; Ayyanar, R.: 2007. A Cosimulation Approach to Model-Based Design for Complex Power Electronics and Digital Control Systems. Arizona State University. Tempe. USA. SCSC.
- Rafull, L. Z. L.; Queiroz, D, M.; Souza, C M. A.; Pinto, F. A. C. 2006. Modeling and analysis of an automatic control system for harvesting machine cutting height. *Rev. bras. eng. agríc. ambient.* , vol.10:751-758.
- Reynolds, P.F., 1988. Heterogeneous distributed simulation. *Proceedings of the 1988 Winter Simulation Conference* p.206-209.

ACKNOWLEDGMENTS

We acknowledge the financial support provided from PIIC/URI.

AUTHORS BIOGRAPHY

Tales Marchesan Chaves is a Computer Science undergraduate student at URI, Santo Ângelo, Brazil. His research interests include simulation and embedded devices.

Braulio Adriano de Mello is a professor in the Computer Science Department of the Lavras Federal University (UFLA), Lavras, Brazil. His research interests include systems modeling and heterogeneous simulation. Mello has a BS in computer science from the Passo Fundo University and a PhD in computer science from UFRGS. He is a member of the Brazilian Computer Society (SBC).

Paulo Betencourt is a professor in the Engineering and Computer Science Department of URI University, Santo Ângelo, Brazil. His research topics are focused on software engineering.

SIMULATION-BASED MODEL OF A DISTRIBUTED GMPLS-BASED IP-OVER-OPTICAL NETWORK

S. Albarrak

Department of Information Technology and Communication,

King Fahad Security College, Saudi Arabia

barraks@kfsc.edu.sa

ABSTRACT

IP-over-optical networks have been identified as promising candidates to underpin future telecommunication networks. Such a structure rely on an intelligent control plane, built based on the GMPLS protocols, that can establish and tear down lightpaths on-demand and provide quality of service (QoS). This motivates both industry and research work into the convergence of IP and optical network.

This paper presents in detail a simulation-based model for a distributed GMPLS-based IP-over-optical network using the OMNeT++ platform. This work ensured that such an implementation mirrored as far as possible the operation and performance of real multi-layer networks. Therefore, this model can be considered as a basis for both industry and research to investigate key issues that affect the integration of IP and optical layers.

Keywords: GMPLS, Distributed system modeling and simulation, IP-over-optical network.

1. INTRODUCTION

In recent years, the research and industrial communities have increased their efforts into the development of optical network technologies, at both the physical and management layers, in order to progress beyond point-to-point transmission systems and proceed to all-optical networks. Consequently, the Optical Transport Network (OTN) structure with its own data plane and control plane has been introduced. Generalised multi-protocol label switching (GMPLS) has been developed by the Internet Engineering Task Force (IETF) to form an intelligent control plane for the OTN. This control plane is able to establish and tear down lightpaths on-demand and to provide quality of service (QoS) (Bernstein, et al. 2004; Tomsu and Schmutzer 2002). As a result, a two-layer approach, consisting of the IP layer built directly over the optical layer, has been identified as a promising candidate for the structure of next generation networks (NGN). Such a structure has its own control and data plane. The control plane can be an overlay or peer model based on the GMPLS protocols while the data plane consists three overlaid topologies; link, lightpath, and label switching path (LSP) topology. Such a structure is still under development and there are many subjects worth considering in detail, practically, under distributed behaviour (Marchese 2007).

Consequently, it is essential to provide an efficient and scalable IP-over-optical network model to support such investigation required by both industry and research. However, most of the previous studies, investigate GMPLS-based IP-over-optical network, describe such a model implicitly whereas the focus was given to the subject and performance results (Wang, et al. 2002). Consequently and in respect of their efforts, this paper considers explicitly this issue and present a distributed GMPLS-based IP-over-optical network using the OMNeT++ platform. Such a model should mirror as far as possible the operation and performance of real multi-layer networks.

In general, there are two options when investigating the performance of IP-over-optical networks: mathematical models and simulation models. The former uses mathematical techniques to implement communication networks with certain numerical parameters. The latter uses a computer to simulate real networks in terms of structures and operations. Moreover, simulation models are used to implement models of networks which are complex, where such models are difficult to study mathematically. In some circumstances like model validation, it is desirable to use both simulation and simple cases of mathematical model to test or validate the model functions. Consequently, a simulation model is used throughout this work, as the network structure being represented is complex.

Therefore, two approaches can be used to develop a simulation model; in-house or using simulation packages. Such packages are very efficient in terms of saving time, developing, debugging and executing the simulation model. Several well-known packages are available to use such as OMNeT++, NS2 and OPNET. A full comparison between such packages is presented by Begg, et al. (2006). This work adopts the OMNeT++ simulator as simulation platform for a number of useful features as explained in Section 2.

This paper begins by providing an overview of the OMNeT++ simulator. Sections 3 describe constituent elements comprising the network model. The model functionality is explained in Section 4. In Section 5, a number of preliminary simulation-based experiments are presented. The prime aim of these experiments is to validate model behaviour and assumptions. Finally, the paper is concluded in Section 6.

2. OMNET++ SIMULATOR OVERVIEW

This work uses the OMNeT++ (Objective Modular Network Testbed in C++) simulator as discrete-event simulation software platform. OMNeT++ is a public-source simulator with a generic and flexible architecture designed to simulate computer networks, multi-processors and other distributed systems (Varga 2006). Internally, OMNeT++ is an object-oriented modular test-bed, whereby each module in the network is implemented as an object. Thus, it supports hierarchically nested modules as illustrated in Figure 1a. Simple models are the lowest level of the hierarchical structure where any simple model is associated with C++ file which describes its behaviour. In compound modules used for grouping or aggregating simple modules, therefore, there are no active behaviours associated. The highest level of the hierarchical structure is called a network model. These models communicate with each other using messages passed through channels. These messages can be used to model a number of entities: events, data packets, frames, cells, bits, or control messages and so on. Moreover, OMNeT++ provides a flexible procedure for initialising and passing module parameters as illustrated in Figure 1b. A text file called omnet.ini is utilised for this purpose.

Besides hierarchical modules and parameter initialisation features, the OMNeT++ simulator offers a number of useful features such as class library, graphical network editing and animation, data collection, graphical presentation of simulation data, and random number generators. Moreover OMNeT++ simulator works under different operating systems with good manuals (Varga 2006).

3. MODEL STRUCTURE

The structure consists of a set of nodes connected by a set of paired fibre links. Internally, each node consists of an edge router connected to an optical cross connection (OXC) as shown in Figure 2. The router and OXC may be placed in a separate location or they may be combined to form a single node with a common control plane. The network structure can be seen from

two angles: a data plane and a control plane. The data plane is an overlay model with three topologies: link, lightpath, and Label Switched Path (LSP) topology. On the other hand, The control plane supports routing and discovery protocols, signalling protocols, survivability mechanisms, and information databases. In practice, the control plane is implemented using software, and hence the effectiveness of such a control plane requires full compatibility between all of these software protocols.

The edge router control plane has a similar structure to the OXC control plane in terms of the required protocols and databases. However, in order to satisfy smart-edge simple-core network structure, the internal structure, implementation and functions of protocols are quite different. The switching part of the edge router has two sections: an optical section and an electrical section. The former is responsible for the transfer of signals between the optical and the electrical domain. The latter is responsible for the smart edge functions undertaken in the electrical domain.

The data plane is essentially a hardware implementation. It is responsible for switching incoming wavelength traffic to the same or a different output wavelength, according to the information presented in the wavelength routing table and wavelength converting availability. Wavelength demultiplexing and multiplexing is required before and after wavelength switching.

each wavelength coming from the OXC local port is connected to a router port, whereby the number of wavelengths that can be handled is based on the number of transceivers equipped in the router. Thus, the number of bidirectional lightpaths that can be simultaneously established by any router is equal (at most) to the number of transceivers. Two types of LSPs can be associated with an edge router: LSPs that originate/terminate on the router and LSPs that travel through the router. These paths are represented as variable-bandwidth-guaranteed paths, while a lightpath is represented as a discrete-guaranteed path. Thus, multiple LSPs may be aggregated in a single lightpath.

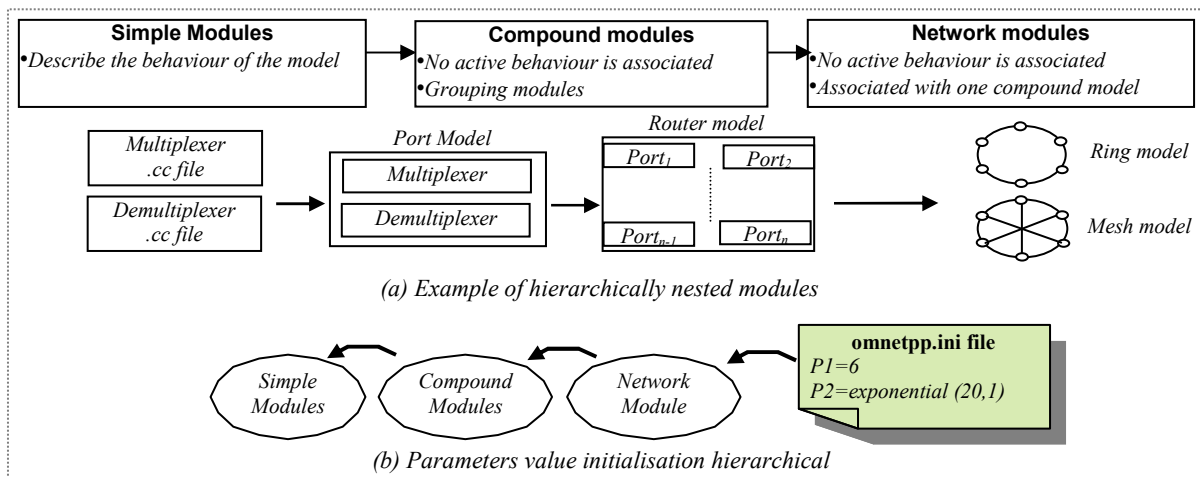


Figure 1: Example of OMNeT++ hierarchically nested modules.

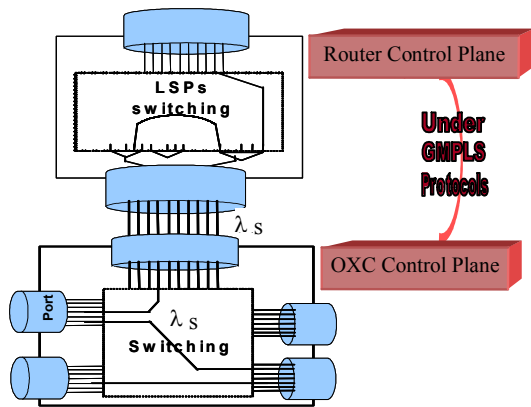


Figure 2: Network node structure.

3.1. OXC Model Implementation

Figure 3 illustrates an implemented OXC model and its associated parameters. Values of the parameters are passed to each node in the configuration stage via text files. The control plane consists of three units: the signalling, the routing, and the recovery unit. These units work independently and communicate with each other using internal messages designed for this purpose. Since the GMPLS protocols are not pre-defined within OMNeT++ simulator, the functionalities of the signalling and routing units are implemented from scratch.

The implemented WR_table unit is responsible for implementing the functionality of the data plane, including connection configurations and wavelength-routing-table updating. This unit is triggered by receiving messages from other units, asking for a particular event, such as establishing a new connection, erasing an existing connection, updating information or checking wavelength states.

Each OXC has a controller model in charge of controlling and scheduling all required functions in the

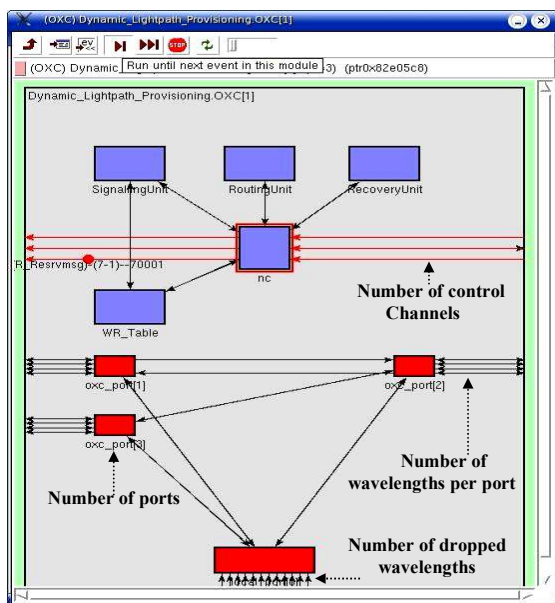


Figure 3: The internal structure of the implemented OXC model.

node. Moreover, it is responsible for delivering messages to remote nodes through dedicated control channels, using GMPLS standard protocol messages implemented by the cMessage class. These messages could be generated locally by the signalling unit or passed over to other nodes.

OXC units require particular information in order to implement their functionality efficiently. Furthermore, the amount of information significantly impacts on model scalability. Four data tables were used in the OXC implementation:

- Wavelength routing table: contains the information that describes the status of wavelengths at each port. This information supports the OXC control plane to make a decision as to whether a required connection can be established or not.
- Lightpath information table: maintains the information about all lightpaths generated from, terminated on, or passed through the corresponding OXC. In particular, the lightpath route information is maintained here. This information enables the control plane to change or modify the lightpath route in order to improve network performance or to recover from failures.
- Network physical topology table: contains the information about the entire network link connectivity. This information enables the routing unit to calculate the appropriate route for a new request. In practice, OSPF-TE routing protocol is responsible for collecting table entries for each OXC. However, this model does not consider this issue. This table is built automatically using functionality provided by OMNeT++ and is updated through routing and signalling protocols.
- Control channel topology table: contains the information that describes the control channel connectivity. The main reason for creating this table is the GMPLS requirement that control messages be exchanged via control channels independent from data plane.

3.2. Edge Router Model

Figure 4 illustrates the implemented router model. The control plane is implemented using GMPLS standards. Each edge router is connected directly to one OXC with the assumption that there is a dedicated control channel between them. Thus, routers can use this channel to communicate with their associated OXC and other routers.

Each router has N ports, and each port is connected to a particular wavelength. Internally, each port has two sub-models: a multiplexer and a de-multiplexer. The former is responsible for multiplexing LSP traffic in the wavelength, while the latter is in charge of de-multiplexing LSP packets from the wavelength. In order to perform these functions, multiplexers and de-multiplexers can communicate with the forwarding-table through messages, to determine the outcome label and the destination port.

Each router has a controller model in charge of controlling and scheduling all required functions within

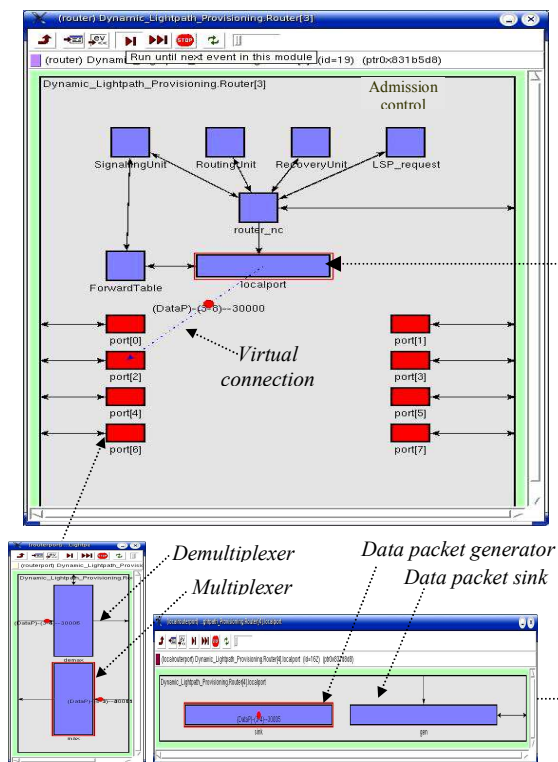


Figure 4: The implemented router model structure with the associated sub-models.

the edge router. Moreover, an admission control model is embedded to manage and handle LSP requests at routers. The structure and functions of such admission control is explained in Section 4.2. A data traffic generator is responsible for generating data packets for each LSP based on the LSP capacity. Such a generator is used in this stage to send a data packet over each LSP for model testing purposes.

Like OXCs, routers require particular information to efficiently implement their functionality. Three tables are described:

- Forwarding table: It contains the forwarding information including in_port, in_label, out_port, out_label and the identification of LSPs. This information supports the data plane in forwarding packets and supports the control plane in notifying the appropriate responsible nodes when failures are detected.
- LSP information table: maintains the information about all LSPs generated from, terminating on, or passing through the corresponding router. This information supports the control plane in changing or modifying the LSP route in order to improve network performance or to recover from failures.
- Logical topology: This table contains the information regarding the existing lightpaths including the type, source, destination, available capacity, routing and SRLG of lightpaths. This information supports the routing unit by calculating the appropriate route for a new request. Typically, such tables are maintained at each router and updated through routing and signalling protocols. Such updating requires time to converge.

However, in this model, for simplicity, it is assumed that routers retrieve such information from a global file which has up-to-date information. Note that, when the convergence time is considered, the available routing information will not precisely represent the current status of data plane, hence this may affect the routing protocol efficiency.

3.3. Link Structure and Implementation

Fibre links are unidirectional links used to connect the network equipment together. In order to provide bidirectional links, a pair of physical fibre links is required. Such bidirectional links support a number of wavelengths, where each wavelength can carry a specific amount of traffic. Figure 5 illustrates the link hierarchy based on the switching granularity. This work supposes that the control plane topology can differ from the data plane topology, and therefore, two types of links are present: data and control links. The former is implemented as a paired fibre with multiple bidirectional connections where each connection characterises a bidirectional wavelength. The latter is implemented as a dedicated channel with explicit capacity. In a real implementation, such a channel can be a dedicated wavelength with the fibre, a timeslot on a TDM link, or a separated link.

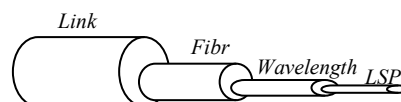


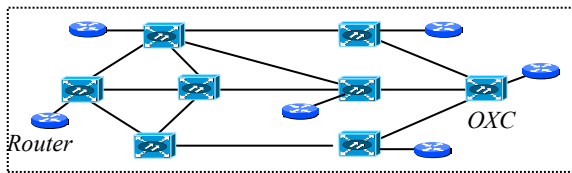
Figure 5: Link structure.

3.4. Network Model Structure and Implementation

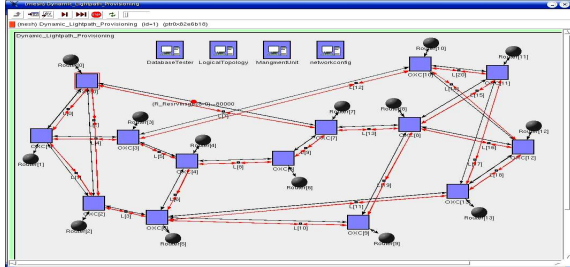
The network model used in this work is a core network consisting of a set of nodes connected by a set of paired fibre links to form a mesh topology, as illustrated in Figure 6a. In this model, the OXC and edge router are implemented to form a single node; therefore, there is no transmission delay considered within the node. On the other hand, in order to deliver data and control messages between nodes successfully, a unique address is specified for each node. Figure 6b illustrates an example of the network connectivity using OMNeT++. A dedicated channel in each link comprises the control plane topology. For simplicity, it is assumed that the control plane topology is identical to the physical link topology. Therefore, a control message may pass several nodes before reaching its destination.

4. MODEL FUNCTIONALITY

This Section summarises the functionality of the simulation model employed to investigate the performance of IP-over-optical networks. Figure 7 demonstrates the general simulation cycle flowchart which consists of several stages: model initialisation, traffic generation, admission control, connection provisioning and performance measurement. The functionality of these stages is explained in detail within the following sub-sections. As shown in the flowchart, the traffic generation stage operates as a separate



(a): Network model structure.



(b): An example of the network implementation: NSFNet.

Figure 6: network implementation

process, independently of the provisioning stage. Since it is assumed that the model operates on a distributed manner, thus, multiple requests are initiated simultaneously at edge routers regardless of the provisioning process's status. On the other hand, it is assumed that the warm-up criteria control the beginning of the performance measurement process. Basically, simulation is initially run for an initial period to put the model into steady state. During this period, requests are generated and processed but no performance statistics are recorded. The simulation is terminated when the simulation-time is finished and also when there is no active provision.

4.1. Traffic Generation

In real networks which operate in a distributed manner, edge routers are responsible for managing their traffic independently based on the on-hand network status (connectivity and traffic distribution). Such

management includes the traffic engineering, connection recovery and lightpath provisioning or teardown. In the model used in this work, a centralised traffic generator is implemented to generate and distribute traffic between edge routers. The generated traffic is represented as an LSP request. In the absence of real traffic information, LSP connections are requested and terminated randomly whereas the requests arrival rate is obtained based on Poisson process and the request holding time is calculated based on negative exponential. Such choices are simple to model and have been applied in the majority of the experiments of circuit switching networks (Wang, et al. 2002; Oki, et al. 2005; Albarrak and Harle 2006). The offered load indicates the traffic load expressed in Erlangs, defined as the product of mean arrival rate per unit time and mean connection holding time. The LSP parameters include the source, the destination, the capacity and the service class, selected randomly based on a uniform distribution. As shown in Figure 7, it is assumed that the traffic generator operates completely independently of other model functionality with its own random seed.

4.2. Admission Control

One of the key concerns in the next generation network is the traffic grooming problem. This problem considers switching the IP low-speed connections called Label Switching Paths (LSPs) onto a high-speed optical connection (lightpath). The lightpath and LSP exhibit a degree of similarity in terms of the provisioning procedure and providing an end-to-end path. They are, however, diverse in terms of granularity in that lightpath granularity is coarser than LSP granularity. Moreover, an LSP may traverse more than one lightpath.

Recent studies describe work on the grooming problem in order to achieve efficient resource utilisation. Two

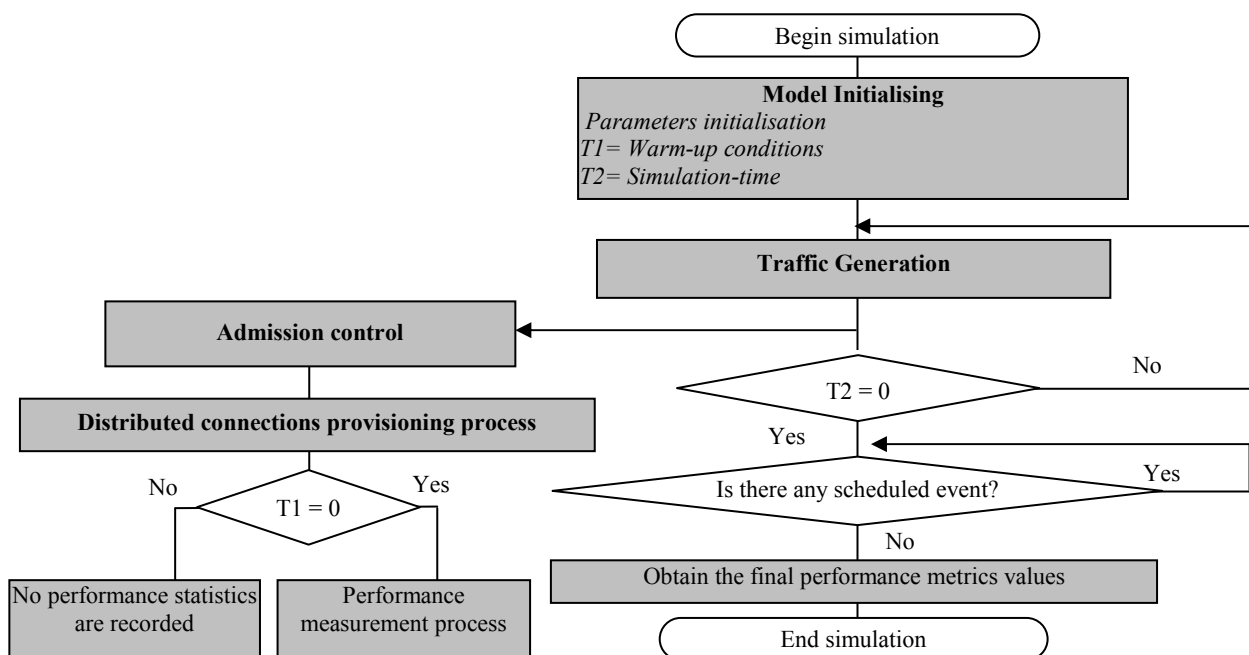


Figure 7: Simulation cycle flowchart.

policies were introduced by the authors of Oki, et al. (2005): *lightpath-create-first* and *lightpath-create-last*. In the former, it is the edge routers that attempt to provision a new lightpath to accommodate new LSP requests. On each lightpath setup failure, the routers attempt to find a route within the existing lightpaths. The latter policy operates in the reverse order. This paper adopts and implements the admission control mechanism presented in previous work (Albarak and Harle 2006). The key feature of the proposed mechanism considers the condition when multiple identical requests associated with a source, destination, and class of service arrive simultaneously at an edge router.

4.3. Connection Provisioning Process

It is assumed that the connection provisioning process is underpinned by GMPLS. Both routing and signalling are involved. The former is responsible for calculating an appropriate route, while the latter is in charge of handling the remainder of the provisioning process. These units must acquire particular information in order to implement their functionality efficiently. Such type of information is explained in Sections 3.1 and 3.2 and can be updated either by signalling or routing protocols. The signalling protocol enables updating tables to maintain local information such as wavelength routing, lightpath information, forwarding and LSP information tables. On the other hand, tables that maintain global information, including the link resource availability, and logical topology tables are updated by means of the routing protocol. The following sub-sections describe in detail the functionality and implementation of routing and signalling units respectively.

4.3.1. Routing Protocol Implementation

This model adopts the source-based explicit routing concept whereby the explicit route is calculated at the source nodes; therefore, this route cannot be modified any more. A routing unit at any node operates independently to obtain the explicit route based on two objects: the on-hand information and applied routing algorithm. From the routing calculation perspective, two main tasks are required in order to achieve an efficient route calculation. Firstly, a network structure representation which aims to model the graph structure as a mathematical structure. Secondly, routing calculation algorithms whose function is to obtain appropriate routes.

In this work, the network structure representation task involves two prime topologies: physical-link and lightpath. The former is considered as a static structure. The latter is considered as a dynamic structure in which network connectivity changes with time. In order to represent these topologies mathematically, some of the graph theory features are employed. Based on the graph theory, the basic structure of a graph (G) consists of a set of vertices (V) connected by means of a set of edges (E). Any two vertices are adjacent to each other if, and only if, they are connected by a direct edge. The adjacency relationship is fundamental to representing

network structures mathematically. Such a relationship is realised by means of an adjacency matrix. This matrix characterises the network connectivity using a two-dimensional matrix, hence, any two vertices v_i and v_j are connected if the value at position (i,j) is not equal to zero. In the case of weighted graph, the value at position (i,j) indicates the weight of the edge connecting v_i and v_j . In this model, both topologies, physical-link and lightpath, are considered as a weighted graph. At the physical-link structure, edge-weight values are determined based on the number of free wavelengths in each link (*1/number of free wavelengths*). At the lightpath structure, these values are calculated based on the amount of available capacity in each lightpath (*1/available lightpath capacity*). Moreover, both structures are considered as a multi-graph in which multiple bidirectional links (fibres in the physical structure and lightpaths in the logical structure) can be connected between any two nodes. In this model, the multi-graph structure is presented as a single graph by selecting one link to connect any two nodes. Such a selected link has the highest available capacity of the all the links involved.

Besides the structure representation task, this model adopts the Constraint-based Shortest-Path-First (CSPF) algorithm to achieve the routing calculation. This algorithm provides an efficient method of computing explicit routes of independent requests based on some specific constraints. Basically, the prime constraint considered by the model is the usage of links expressed clearly using edge-weight values and SRLG condition.

In this model, the two tasks are implemented by means of a standard library called Boost Graph Library (BGL) (Jeremy, et al. 2001). This decision was made for the reason that using well-known standard libraries reduces mistakes and saves time compared to the in-house implementation. The BGL supports the two prime tasks required by the model. The BGL utilises the `adjacency_list` class which supports the network structure presentation task. This class contains various functions used to update, modify and access structures. On the other hand, BGL supports different routing algorithms. One such algorithm is the CSPF which is of specific interest for this model.

Using BGL, the functionality of the routing unit begins by selecting the links which have sufficient resource and calculating their weight values, and then uses the `adjacency_list` class to present network connectivity. Finally, the shortest path is obtained using the CSPF algorithm. This procedure is applied in both IP and optical layer routing units.

4.3.2. Signalling Protocol Implementation

This work employs the Destination-Initiated Reservation (DIR) method using the GMPLS signalling protocol in both IP and optical layers. Figure 8 illustrates the flowchart for this method. When the explicit route for a request (lightpath or LSP) is successfully computed, the signalling unit at the source generates a path message. This message is then

forwarded from the source to the destination and collects the resource information on its way. When the destination receives the path message successfully, a reservation message (Resrv message) is forwarded back to the source. All intermediate nodes, including the source and destination, must ensure that the required resources are available when any message is received. The required resource is as signed within the path message process and completely reserved within the reservation message process. In order to establish a bidirectional connection in one attempt, it is assumed that path messages carry the upstream label to the downstream node while reservation messages inform the upstream node about the selected downstream label. The request will be blocked if there are no available resources along its route. If the request is rejected within the path message route, a path error (PathErr) message is forwarded back to the source, while a reservation error (ResrvErr) message is forwarded back to the destination when the request is rejected within the

reservation message route. Once the destination receives a ResrvErr message, it is responsible for generating and sending back a PathErr message along the route toward the source.

The generalised label introduced by GMPLS supports different kinds of label. Consequently, the label identifies the required resource not only logically but also physically. As an example, with WDM switching, the label is used to specify exactly the required physical resource (wavelength) while in the case of packet switching, the label is used to allocate the capacity in an interface logically.

At the IP layer, the label is presented as integer numbers. However, in the model, it is assumed that the label presents as a pool of numbers where each router works independently to obtain its label. The pool size is limited to a certain maximum number according to the 20 bits assigned within the label structure to specify the label values.

At the WDM switching layer, there are several

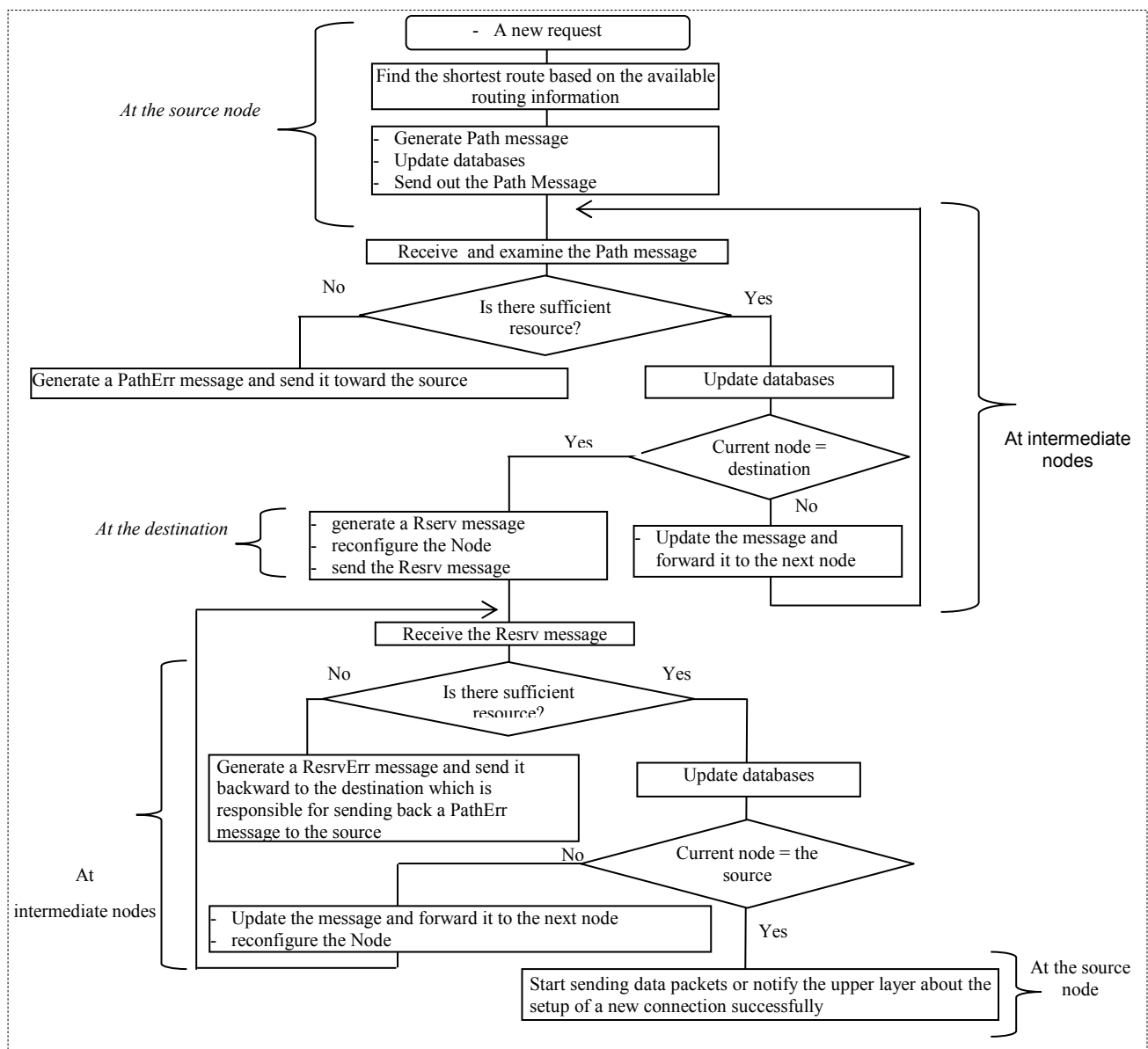


Figure 8: The flowchart of destination-initiated reservation (DIR) method.

strategies which can be used to select a wavelength from a pool of free wavelengths, such as first fit, random, least used, and most used. The strategy considered in this model is first fit. Based on this strategy, the wavelengths are sorted in an appropriate order and the first available wavelength is selected. Figure 9 gives an example of applied wavelength assignment process assuming full wavelength conversion. At source node, the upstream label (λ_5) is selected and a label set object is updated by recording the available labels in its outgoing port (λ_6, λ_{11}). This object is sent within path messages to the next node. When any intermediate node receives a Path message, firstly, an upstream label from the label set object is selected. Secondly, this object is updated by including all available labels in its outgoing port. Finally, the Path message is forwarded to the next node. The same procedure is applied at all intermediate nodes. When the destination receives a path message, it reserves the required resource and sends back a Resrv message. The destination and all intermediate nodes must inform the next node about the selected downstream label in order to reserve the required resource along the route. This procedure is also applicable when the wavelength continuity condition is considered. In this case, the Path and Resrv messages must consider only a specific wavelength during the connection provisioning process. One of the critical issues in a distributed GMPLS-based control plane is contention between messages when multiple provisioning processes begin simultaneously. One solution for such a problem is to apply a retrial

method where, when the connection provisioning process fails, the provisioning process is repeated. Note that, in this model, it is assumed that no repeat behaviour is considered under normal conditions. However, such a model suggests two solutions to partially solve the contention problem: control message prioritisation and Master-slave relationship policies.

The control message prioritisation policy provides different levels of priority based on the type of control messages. Thus, the reservation messages have higher priority than Path messages. This model introduces three status of label reservation: free, soft or hard. Path messages always consider soft reservation, while Resrv messages consider hard reservation. Therefore, the Resrv messages can take advantage of any label not in the hard reservation status.

Master-slave relationship policy solves the contention problem within the same type of messages. This policy relies on the fact that each link connects two nodes. One is set as a master node and another is set as a slave node. Therefore it allows the master node to take higher precedence when the contention occurs. Figure 10 illustrates a simple example. The link between nodes 2 and 7 has one available wavelength and two Path messages x and y are travelling through that link in opposite directions. When the Path message x arrives to node 7, λ_3 is in the soft reservation from connection y . At this time and because node 7 is a slave node for node 2, the connection x will win λ_3 . In the opposite way, connection y will be rejected since there is no wavelength with a free status.

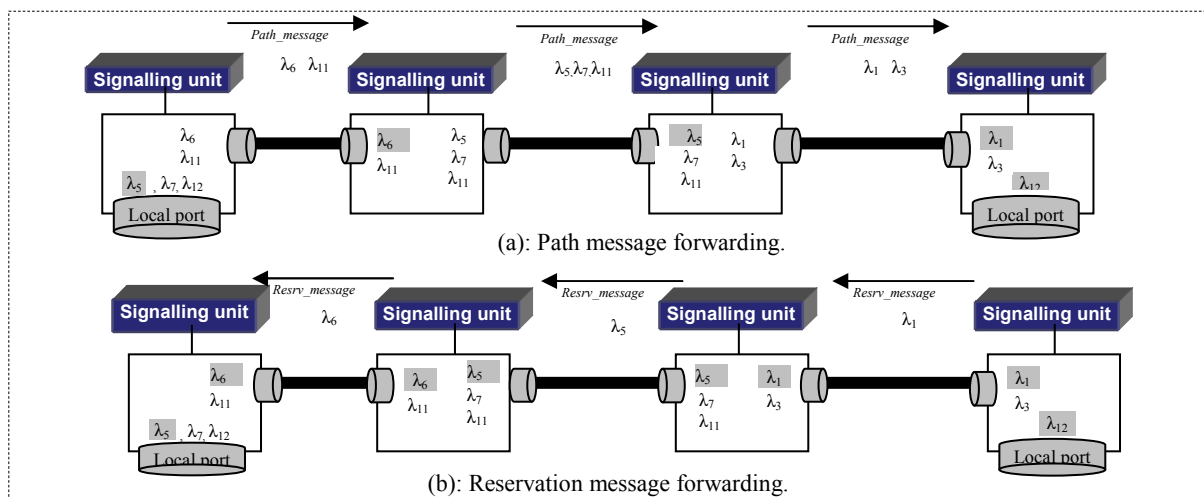


Figure 9: An example of wavelength assignment with full conversion.

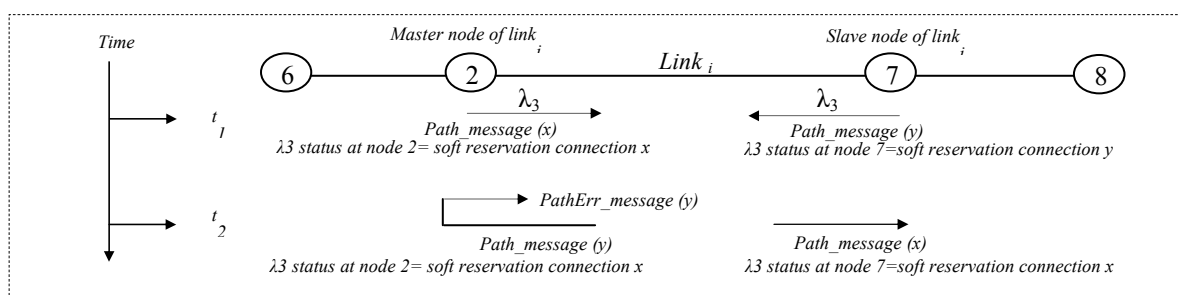


Figure 10: An example of the slave-master relationship.

Moreover, the master-slave relationship is used with first fit strategy to manage the order of available wavelengths between nodes, in particular, within the label set object. As demonstrated in Figure 11, a slave node always records the available wavelengths in the reverse order of a master node.

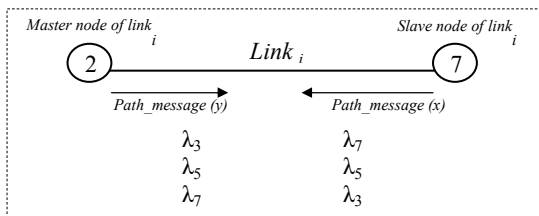


Figure 11: Example of the first fit strategy using slave-master relationship.

4.3.3. Connection Teardown

This model adopts the deletion in progress policy applied in GMPLS. Based on this policy, a Path message with specific objects is sent along the route to inform nodes about tearing down a connection. On the other side of the connection, the actual teardown process is initiated. The actual connection termination procedure begins by sending a teardown message along the connection route. Such a message is responsible for updating the database and reconfiguring all nodes passed through. In the normal operation, teardown can be initiated by either the source or destination node. The deletion in progress policy is necessary, in particular, at the optical layer, the reason being that if an upstream node tore down a connection without previously informing the downstream node, the latter would detect a connection failure.

In this work, it is assumed that LSP is terminated following a generated random time interval after the LSP is established. On the other hand, lightpaths are terminated if they are not being traversed by LSPs. Two steps are followed by routers to teardown a lightpath. Firstly, an unused lightpath is blocked, thus, it is not seen by the routing protocol. Secondly, a teardown request is sent to the OXC to start the formal teardown process.

5. MODEL VALIDATION

The model validation task can be achieved, generally, by considering four model parts: traffic generation, signalling protocol, routing protocol, and the proposed admission control. This section summaries number of preliminary experiments aim to validate the four model parts. Firstly, The traffic generation validation has been achieved by comparing the theoretical and simulation value under different traffic loads. Secondly, the signalling protocol has been validated using two method; *Message Delay* and *control plane database against the data plane connectivity* described in Sections 5.1 and 5.2. Beside the *control plane database against the data plane connectivity* method, a number of simulation-based experiments have been implemented in order to test the routing protocol functionality. four

performance parameters are investigated: network connectivity as shown in Section 5.3, number of wavelengths per link and wavelength capacity, a selected port and lightpath have been tested by monitoring the traffic during the simulation and Erlang B Test . Developed tools are embedded in the model for this purpose. Finally, the proposed admission control is validate by comparing its result with the result conclusion presented by Oki, et al. (2005).

5.1. Signalling Protocol (Message Delay)

In order to analyse the delay time, this model considers three delay components; the link propagation delay, the link transmission delay and the nodal process delay. The link propagation delay is the latency in propagating the bits along the link and it is a function of the link propagation speed and the link length. The link transmission delay is the time needed to transfer/pump data onto a link and is calculated as a function of the link capacity and the message size. The nodal process delay describes the time between the node receiving a message though the input port and the time when the message is sent to the output port, including the time taken to examine a message, to calculate a new route and to perform wavelength switching.

In the following, delay issues will be explained by an example. The topology is presented in Figure 12.a with the assumption that all links are of equal length (1000 km). The process delay is 3ms for the Path and Reservation messages and 1ms for other messages. The OXC reconfiguration time is 1ms. The total delay in establishing two lightpaths (LP1 and LP2) and LSP is obtained by simulation and mathematically, as illustrated in Figure 12.b and 12.c respectively. Mathematically, the delay required for establishing a bidirectional connection is equal to the summation of these components. Based on the hop-by-hop reservation technique explained in Section 3.4.2. The results prove that the delay components implemented in the model are equal to the theoretical values. This experiment express that the signalling protocol implementation is working correctly.

5.2. Control Plane Database and Data Plane Connectivity

The DataBaseTester Model is designed to support database validations by testing the control plane database against the data plane connectivity actually implemented in both IP and optical layers. It is assumed that such a model operates during the simulation to check its integrity. This model directly affects the speed of the simulation; thus, a testing time parameter is suggested to adjust the period between testing events.

From an optical layer database perspective, the Omnet++ simulator presents classes that can be used to discover module parameters and gate connectivity, such as cTopology, sTopoNode, and cGate. Using these classes, it is possible to generate the logical topology independently of the control plane database. The output

of this function is saved in a text file. Subsequently, the tester triggers the comparison functions to check the entities within the text file and the optical layer database tables.

In order to check the IP router database, another technique is applied. The operation is started by triggering all routers to send test data packets through the data plane in both directions along each LSP and, when received on the other side, test packet information is recorded independently in a text file. Thus, this file can be compared with the OSPF-TE databases in each router. The main reason for applying such a scheme is the fact that LSPs are virtual connections.

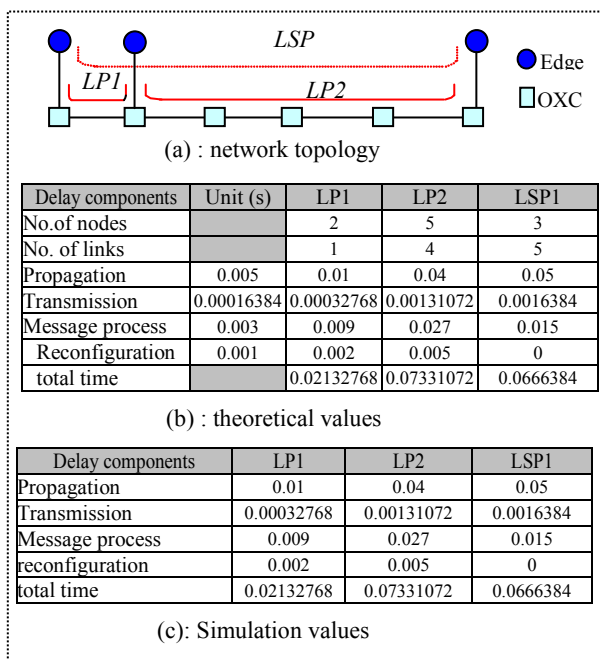


Figure 12: An example of the model and theoretical timing comparison.

5.3. Effect of Network Connectivity

Three network topologies are tested by the model as illustrated in Figure 13 where all links have the same length (1000 km). The performance metric of interest is the blocking probability which gives the ratio of the number of rejected connections over the number of requested connections in the network. Figure 14 shows the obtained blocking probability when different numbers of links are embedded. As expected, the blocking ratio declines as the connectivity is raised.

6. CONCLUSION

This paper explained in detail the simulation model built using OMNET++ considering a distributed GMPLS-based IP-over-optical network model. Internally, the model was described in terms of structure, implementation and functionality. A number of preliminary simulation-based experiments were presented to raise confidence in the model behaviour and assumptions.

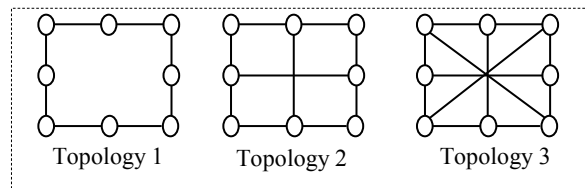


Figure 13: Examples of network topologies.

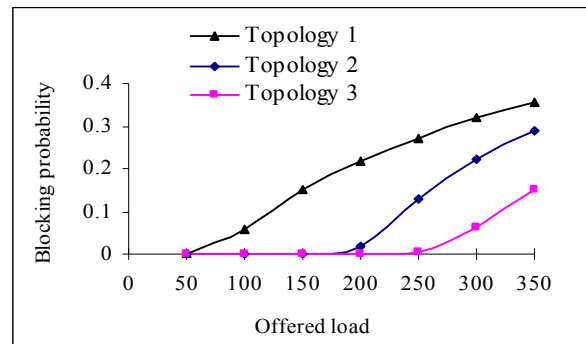


Figure 14: Blocking probability performance comparison between different network topologies.

The implemented model was considered as a basis for investigating key issues that affect the integration of IP and optical layers. These issues are considered as future work of this paper.

REFERENCES

- Albarrak, S., Harle, D., 2006. An Edge Admission Control in a Distributed GMPLS-Based IP/WDM Network. *IV GMPLS Workshop*, 30-31 March, Girona, Spain.
- Begg, L., Liu, W., Pawlikowski, K., Perera, S., Sirisena, H., 2006. *Survey of Simulators of Next Generation Networks for Studying Service Availability and Resilience*. technical report, University of Canterbury, New Zealand.
- Bernstein, G., Saha, D., Rajagopalan, B., 2004. *Optical network control: architecture, protocols, and standards*. Addison-Wesley Professional.
- Jeremy, G., Lie-Quan, L., Andrew, L., 2001. *Boost Graph Library, The: User Guide and Reference Manual*, Addison-Wesley Professional.
- Marchese, M., 2007. *Qos over Heterogeneous Networks*. John Wiley & Sons.
- Oki, E., Shiimoto, K., Shimazaki, D., Yamanaka, N., Imajuku, W., Takigawa, Y., 2005. Dynamic Multilayer Routing Schemes in GMPLS-Based IP+Optical Networks, *IEEE Communications Magazine*, 43, 108-113.
- Tomsu, P., Schmutzer, C., 2002. *Next Generation Optical Networks*. Prentice Hall PTR.
- Varga, A., 2006. *Omnet++ user manual*. OMNeT++ Community Site, Available from: <http://www.omnetpp.org>.
- Wang, J., Sahasrabudhe, L., Mukherjee, B., 2002. Path vs. subpath vs. link restoration for fault management in IP-over-WDM networks: performance comparisons using GMPLS control signalling. *IEEE Communications Magazine*, 40, 80-87.

THE DISCRETE EVENT SIMULATION FRAMEWORK DESMO-J AND ITS APPLICATION TO THE JAVA-BASED SIMULATION OF MOBILE AD HOC NETWORKS

J. Göbel^(a), A.E. Krzesinski^(b), B. Page^(a)

^(a) Department of Informatics, University of Hamburg, 22527 Hamburg, Germany

^(b) Department of Mathematical Sciences, University of Stellenbosch, 7600 Stellenbosch, South Africa

^(a) [goebel.page}@informatik.uni-hamburg.de](mailto:{goebel.page}@informatik.uni-hamburg.de), ^(b) ae1@cs.sun.ac.za

ABSTRACT

This paper presents an introduction to DESMO-J, a comprehensive framework for implementing discrete event simulation models in Java. We describe the most important DESMO-J components. Key features include a clear separation of model development and experiment conduction and the support of both event-oriented and process-oriented model descriptions; both views can be combined within a single model. DESMO-J is extensively used in research, teaching and technology transfer at the University of Hamburg and at other universities. However, applications of DESMO-J have been reported primarily in logistics. We present a simulation model from an entirely different area, namely the simulation of an abstract mobile ad hoc network. A MANET is a collection of mobile devices dynamically forming a wireless network. The model concerns the evaluation of a protocol for the decentralized and fair data rate allocation. The implementation and a typical experiment are described, illustrating the use, capabilities and flexibility of DESMO-J.

Keywords: discrete event simulation, simulation framework, Java, mobile ad hoc networks

1. INTRODUCTION

This paper presents an introduction to DESMO-J (**D**iscrete **E**vent **S**imulation **M**odelling in **J**ava), a comprehensive framework developed at the University of Hamburg, Germany for rapidly implementing discrete event simulation models in the widely-used object-oriented programming language Java. We first describe DESMO-J's black box components which provide the simulation infrastructure and the hot spot components which are abstract classes to be implemented by the user such as events and processes which determine the behaviour of the simulator. We then describe the important design features of DESMO-J.

We next present an application of DESMO-J to model the behaviour of mobile ad hoc networks (MANETs) which are networks of wireless devices which do not necessarily require a communications

infrastructure. In this way the reader gains an impression of how simulation modelling in DESMO-J is conducted. The abstract MANET model is based on Crowcroft, Gibbens, Kelly and Östring (2004); the focus of the MANET model is the design of a decentralized and fair protocol to allocate resources (battery power and bandwidth) such that nodes are provided with incentives to use their resources to relay transmissions on behalf of other nodes.

The paper is organized as follows. Section 2 provides an introduction to the DESMO-J framework, focusing on its most important features, components and design decisions. Section 3 describes MANETs as an application area. A protocol for the decentralized and fair allocation of resources (bandwidth and battery power) which determines the data rates based on congestion pricing is described. This section also establishes the major features that a simulator requires for the purpose of simulating the abstract MANET model. Section 4 shows how the MANET environment is implemented in DESMO-J; this section is completed by describing a typical MANET simulation experiment and discussing the results. Conclusions are given in Section 5.

2. DESMO-J

DESMO-J is a framework in Java developed at the University of Hamburg for rapidly implementing discrete event simulation models. DESMO-J is licensed under the Apache license version 2.0. The following sections provide a short introduction to DESMO-J covering some general properties, important components and simulation modelling as well as some design decisions. The reader is referred to Page and Kreutzer (2005) and to the web page <http://www.desmoj.de> from where the binaries, the source code and other resources can be obtained. These resources include an example model, the API documentation and a tutorial; the latter eases getting started with simulation modelling in DESMO-J: it contains a step-by-step implementation of a small container terminal model using either the event-oriented or the process-oriented view.

2.1. General

A framework can be defined as a collection of different components with coherent software architecture and a defined cooperative behaviour serving a particular task. As a framework at the level of a general purpose programming language, DESMO-J requires the user to code discrete event simulation models in Java. Although simulators at the tool or model level (see Page and Kreutzer 2005, Chapter 9.1) may provide a quicker means of developing standard models such as queueing/server systems, the flexible approach adopted by DESMO-J ensures that complex systems can be adequately mapped into a simulation model as no simulator-specific programming languages or restricted APIs constrain model implementation. This facilitates the modelling of sophisticated strategies and complex system behaviour.

The advantages of choosing Java also include addressing a wide-spread community of programmers, clear and robust object-oriented programming without pitfalls like dangling pointers and memory leaks, as well as platform independence. At the same time, the potential disadvantages of Java, most notably that run-time performance is worse in terms of execution efficiency than for example C++, have been lessened by improvements in just-in-time compilation and the optimization of garbage collection.

Several example applications of DESMO-J have been published including multi-agent city couriers (Knaak, Meyer and Page 2004), harbour logistics (Page and Neufeld 2003), vehicular traffic (Wittmann, Göbel, Möller and Schroer 2007) and material flow analysis (Wohlgemuth, Page, Mäusbacher and Staudt-Fischbach 2004).

The clear design and the ease and flexibility of simulation modelling provided by DESMO-J have resulted in this simulator being intensively used in simulation courses for students of Computer Science and Information Systems at the University of Hamburg and elsewhere, in applied research projects as well as for industrial simulation studies in technology transfer projects (Page and Kreutzer 2006). The business process modelling tool *iProcess Suite* (TIBCO Software Inc., <http://www.tibco.com>) relies on DESMO-J as a simulation engine.

2.2. Components and Modelling

DESMO-J provides two groups of components (Java classes). *Black box components* are “closed” classes that are ready to use: they provide a complete environment to execute discrete event simulations. This environment includes the core simulation infrastructure consisting of an event list storing pending events and processes, the scheduler controlling model execution, the simulation clock as well as an **Experiment** class to integrate these components. The environment also includes facilities for collecting and reporting statistical data, random number generation, a variety of probability distributions as well as generic model components like queues.

Given the availability of ready-to-use black boxes, the user can concentrate on implementing the properties and the behaviour of the system being modelled. For this purpose, DESMO-J provides a second class of components referred to as *white box components* or *hot spots*. White box components are abstract Java classes whose implementation has to be completed by the user.

The DESMO-J user can choose between an event-oriented view and a process-oriented view of describing a model. The event-oriented perspective (“bird’s eye view”) requires implementing subclasses of the hot spot **Event** defining the changes that are to be applied to or triggered by an entity at a certain instant in time. Similarly, subclasses of **ExternaleEvent** represent state changes that are not associated with a specific entity, but with the model as whole, for instance external arrivals. Entities are objects which are derived from the class **Entity**. Examples of possible state changes conducted by events and external events in their so-called event routines are: attributes modified, entities enqueued or dequeued, events scheduled into the event list for later execution, or events in the event list being brought forward or postponed (rescheduled) or cancelled.

In contrast, a process-oriented model definition involves describing the activities conducted by every entity over time during its life cycle; this yields a “worm’s eye view” representing the model behavior from the perspective of the entities. Process life cycle definitions require subclassing **SimProcess**. In addition to modifying attributes and queueing operations, a life cycle may include waiting for a fixed amount of time or for an indeterminate period until activation by another process, and activating or interrupting other processes. Note that events are executed at specific instants in time without interruption while processes (potentially) persist as time passes, handing over program control amongst each other.

An important feature of DESMO-J is that both views can be combined in a single model, for example an event activating a process that in turn schedules or cancels another event. The user is free to use whichever modelling view is better suited for certain model entities and their behaviour.

Finally, regardless of the modelling view used for model description, any model requires a subclass of the hot spot **Model** in which all model components are created, and dynamic behaviour is triggered by scheduling initial events or activating processes.

2.3. Design Decisions

Fig. 1 shows the hierarchy of some important DESMO-J classes. **NamedObject** (implicitly derived from **java.lang.Object**) as the root for all DESMO-J classes ensures that all objects in DESMO-J have a meaningful name. The separation of the model, which is composed from objects derived from **ModelComponent**, and the experiment (according to the *Facade* design pattern, the class **Experiment** initializes and manages the simulation infrastructure)

keeps model development and experimentation independent of each other. This for example allows for a series of experiments, where each experiment is represented by different **Experiment** objects, to be conducted with the same model **Model** object.

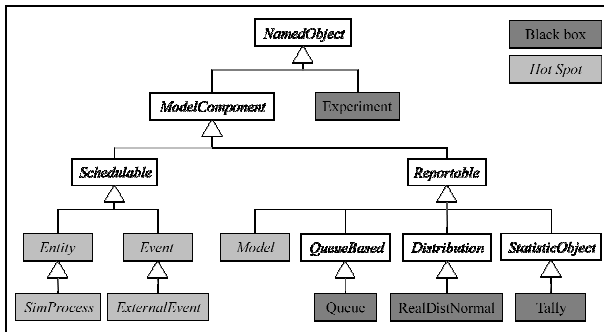


Figure 1: The DESMO-J class hierarchy

Model components are partitioned into two groups. First, components can be derived from **Schedulable**, representing objects which the discrete event scheduler can process yielding the desired model behaviour. Note that **SimProcess** is derived from **Entity**, allowing for scheduling events to occur for a given entity as well as for a process, which facilitates combining event- and process-oriented models. Second, static model components are derived from **Reportable**, denoting their ability to report statistical data generated by an experiment, for example a **Queue** reporting its average length and waiting time. DESMO-J also emphasizes the encapsulation and re-usability of model components: a **Model** class, itself derived from **ModelComponent**, can be used within another model, thus forming hierarchical structures of submodels.

Other features and design decisions of DESMO-J include data being collected automatically if desired (based on the *Observer* design pattern), a GUI for model and experiment parameterization and experiment execution as well as white- and black box components for higher-level process synchronization such as processes waiting for resources from a pool which are acquired/released mutually exclusively.

3. MOBILE AD HOC NETWORKS

The applications of DESMO-J already published as quoted in Section 2.1 may give rise to the impression that DESMO-J is primarily used in logistics simulations. However, logistics is by far not the only field where discrete event simulation can be successfully applied. This section presents mobile ad hoc networks (MANETs) as an entirely different application area. The focus of the simulation is to evaluate a protocol for resource (battery power and bandwidth) and data rate allocation that is decentralized and fair. The protocol description was initially presented in Göbel and Krzesinski (2008b). The section concludes by establishing the requirements a simulator has to meet in order to adequately model the abstract

MANET environment based on Crowcroft, Gibbens, Kelly and Östring (2004).

3.1. Definition

Mobile ad hoc networks (MANETs) are formed by mobile nodes using wireless communication. Each node can be source and a destination of data transmissions and also functions as a router. This cooperation ensures that nodes that are out of each other's radio range can transmit to each other without relying on an existing infrastructure or on the network topology being known in advance. A survey of MANETs focusing on Quality of Service aspects is given in Reddy, Karthigeyan, Manoj and Murthy (2006). A typical MANET application might be passengers at an airport terminal carrying wireless-enabled devices which allow multi-hop routes to be established to connect these devices to each other or to an Internet access point in case the transmission range does not suffice to directly connect to the target.

3.2. Resource Allocation and Fairness

Suppose the nodes that form a MANET are owned by a single authority. Then it is in the authority's interest that the nodes make their resources (battery power and bandwidth) available to forward data transmissions between origin and destination nodes that are not in direct radio contact with each other and which are dependent on one or more relays to forward their transmissions. However, if the nodes belong to different authorities, then a relay might lack an incentive to drain its battery power and allocating its bandwidth by altruistically supporting data transmissions of which it is neither the source nor the destination. The functioning of a MANET depends on the participation of the nodes. However, a node will only participate if *fairness* in resource allocation is enforced: if a node acts as a relay (transit) node and grants its resources to enable the data transmissions of other nodes, then the relay node will expect the same service in return when it attempts to send its own transmissions on multi-hop routes. Moreover, should a node never need to use another node's resources, then that node will demand monetary compensation for the use of its resources.

Crowcroft, Gibbens, Kelly and Östring (2004) investigate an abstract fluid-level MANET model, for which they present a mechanism of resource pricing and data rate control based on congestion prices that are assigned to the resources at every node. Prices are measured in terms of a virtual currency referred to as credits; the power congestion price $\mu_j^p(t)$ that node j charges expressed in credits per unit power (credits/W) satisfies the differential equation (DE)

$$\frac{d}{dt} \mu_j^p(t) = \frac{\kappa \mu_j^p(t)}{\Gamma_j} (\gamma_j(t) - \Gamma_j) \quad (1)$$

with initial value $\mu_j^p(0) = 1$ where κ is a constant of dimension seconds⁻¹, $\gamma_j(t)$ is the power in use at

node j at time t and Γ_j is target power node j is willing to make available to the network. Likewise, the bandwidth congestion price $\mu_j^B(t)$ expressed in credits per unit flow (credits/Mb) satisfies the DE

$$\frac{d}{dt} \mu_j^B(t) = \frac{\kappa \mu_j^B(t)}{C_j} (c_j(t) - C_j) \quad (2)$$

with initial value $\mu_j^B(0) = 1$ where $c_j(t)$ is the bandwidth in use at node j at time t and C_j is the target bandwidth of node j not to be exceeded by the MANET traffic.

These DEs leave the congestion prices constant when the current utilization of a resource matches the target utilization. The prices increase when the resources are over-utilized and the prices decrease when the resources are under-utilized. The DEs must be evaluated frequently to ensure the power $\gamma_j(t)$ and bandwidth in use $c_j(t)$ will only temporarily exceed the target resource utilization: Over-utilization will cause the congestion prices to increase, which will lower the flow rates. Crowcroft, Gibbens, Kelly and Östring (2004) suggest numerically evaluating the DEs every $\Delta = 0.01$ seconds which will result in target resource utilization being exceeded infrequently and by small amounts.

Each node has a credit balance $b_s(t)$. Data transmissions are connected on the least-cost route, assuming a route to the destination exists. During the data transmission, the originating node pays credits for the power and bandwidth congestion costs incurred at the downstream nodes along its route according to prices determined by the DEs described above. In return, the source node receives credits for data transmissions that it relays and for which it provides its resources for transmitting or receiving. During each second the nodes are allowed to spend a fraction $\alpha b_s(t)$ of their current credit balance $b_s(t)$ to pay for the resources that they will use at the downstream nodes along their route. The data rate allocated to an originating node will therefore increase/decrease as its credit balance increases/decreases. A data transmission terminates after the desired amount of data has been sent. A transmission is discarded if it has not completed after a maximum transmission duration.

The scheme provides the required incentive to collaborate when the nodes do not belong to a single authority: the nodes need to acquire credits by providing their resources to other nodes in order to be able to send traffic themselves. However, some nodes may not be able to attract sufficient transit traffic to cover the costs of transmitting their own traffic. A node at the edge of the network may hardly be used as transit at all. In contrast, a node at the centre of the network may face such a high demand to act as a relay that it earns more credits than it is able to spend.

This results in a flow of credits from (1a) those nodes that are located in disadvantageous positions and from (1b) those nodes that transmit more data than the

average node, to (2a) those nodes whose credit balance benefits from a favourable location and to (2b) those nodes with low amounts of data to transmit. *Credit discounting* was therefore suggested in Crowcroft, Gibbens, Kelly and Östring (2004) allowing nodes whose credit balance is less than a certain amount (the target credit balance) to create a certain amount of credits per unit of time, while those nodes whose credit balance exceeds the target balance must destroy surplus credits at the same rate. Since the creation and destruction of credits improves or degrades the amount of resources a node can obtain subsequently, compensation is required: As proposed in detail in Göbel and Krzesinski (2008b), nodes creating credits are monetarily charged (e.g. a certain amount of dollars per credit). The payments are deposited in a fund which is used to compensate those nodes which destroy credits. Note that evaluating net payments provide a means of monetarily evaluating node locations.

3.3. Local Control

The performance of the abstract MANET model proposed by Crowcroft, Gibbens, Kelly and Östring (2004) depends on the following factors (not necessarily exhaustive)

- propagating pricing information subject to delays
- parameters of the data rate allocation scheme, most important the rate of creating/destroying credits at under/overprovisioned nodes
- assigning different amounts of energy and bandwidth to the nodes
- various types of node movements (see Camp, Boleng and Davies 2002 for an overview)
- radio interference (investigated in Göbel, Krzesinski and Mandjes 2007).

Crowcroft, Gibbens, Kelly and Östring (2004) demonstrate that in the absence of radio interference their mechanism of resource pricing and data rate control achieves optimal resource utilization (and near maximum data rates subject to the constraints is resources the node are willing to provide to the network) and global stability of the system.

However, although no central controller is needed, their scheme assumes that the resource utilization and the prices at each node are instantaneously available at all nodes in the network. For that reason, their results represent an upper bound on the performance of a fluid-flow network where data rates are determined based on (partially) outdated prices that have to be propagated from node to node along each route subject to delays in order make them available locally.

An example of a signalling protocol is presented in Göbel and Krzesinski (2008b) to conduct this exchange of information. Two types of signals are introduced:

- An *update cost (UC) signal* is sent periodically from the destination to the origin of the data transmission: this signal gathers the local resource prices which are valid at the instant the signal arrives at each node on the route
- Once the update cost signal reaches the origin, the source updates the data rate of the transmission as well as its credit balance and returns an *update flow (UF) signal* to the destination: this signal disburses the credits to be paid to each transit node on the multi-hop route for the data transmission during the past period.

Special UF signals announcing a preliminary data rate of 0 are sent to initialize a flow or to reroute a flow (rerouting will occur if a relay node along a multi-hop route leaves the network because its battery power is exhausted) and to terminate a data transmission. These signals do not represent a reply to an UC signal. Fig. 2 summarizes the protocol – see Göbel and Krzesinski (2008b) for more details.

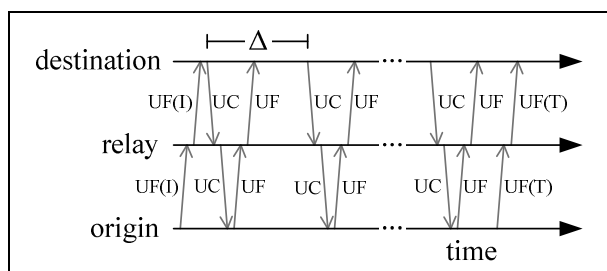


Figure 2: The protocol

3.4. Simulation Requirements

To evaluate the impact of decentralized (i.e. signal-based) data rate allocation on the performance of the decentralized network in terms of data throughput, resource utilization and transmissions discarded, a simulator for the abstract MANET environment is required that must meet the following criteria:

- The simulator must adequately map the abstract MANET system described above into a valid simulation model. To do this, the simulator must be sufficiently flexible: it must not enforce undesired abstraction or idealization by not supporting the simulation of parts of the model, nor must it impose an overhead by requiring unnecessary details to be simulated. This implies that the simulator supports modelling using the event-oriented view (see Section 2.2) since the model behaviour depends upon triggers (for example a data transmission starting, an UC signal processed, a node departing) occurring at certain instants in time. Conversely, there are no complex time consuming activities for which the process-oriented view would be better suited.

- The simulator must support the automated collection, aggregation and reporting of relevant data.
- The simulator must support the separation of the MANET simulation logic and the experiment scenarios such as the spatial configuration of the nodes, resource provisioning at the nodes and the movement patterns of the mobile nodes. This eases making changes to the network topology configuration.
- Modifications and extensions to the model are facilitated thanks to the object-oriented programming paradigm. Modifications like adjusting movement patterns are conducted by refining the relevant component already defined in the simulation model. An example extension is given in (Göbel and Krzesinski 2008a) where the MANET simulator components were augmented to model a mobile sensor network where area coverage currently attained is measured, which the autonomously moving nodes seek to maximize.

DESMO-J meets these criteria. Flexibility, ease of modification and extension are inherited from using Java as a programming language. DESMO-J imposes no restrictions and hardly any overhead to the definition of entities (derived from **Entity**) and behaviours (method **eventRoutine()** of subclasses **Event** and **ExternalEvent**): Since derived from **Named Object**, a name must be assigned to entities and events. Additional overhead may be incurred by including the entities and events in the trace output, or by increased simulation run-time performance, for example each **Schedulable** holding a reference to its current event list entry, if currently scheduled. DESMO-J also provides comprehensive reports which are generated automatically after simulation experiment completion. Different experiment scenarios are described by parsing XML input files which contain the different network configurations.

As far as disadvantages are concerned, we have already mentioned Java's bias in run-time performance (Section 2.1). Moreover, DESMO-J does not (yet) provide a native visualization facility. For the purpose of MANET simulation development, visualization is not strictly necessary, yet it could be useful: although visual impressions are no substitute for empirically well-founded analysis, visual metaphors allow for rapid communication and comprehension of model states and behaviours. In particular, erroneous or implausible behaviour may be easier to detect visually than by analyzing aggregated statistical data (see Page and Kreutzer 2005, Chapter 9.6). In this regard, the DESMO-J user can implement his/her own visualization using Java's built-in graphics libraries or (more conveniently) use existing frameworks serving this purpose; our MANET simulator uses the

framework *JFreeChart* (<http://www.jfree.org/jfreechart>) to visualize mobile node movement.

When comparing a general-purpose simulator like DESMO-J to dedicated telecommunication network simulators like OMNeT++ (<http://www.omnetpp.org>) or ns2 (<http://www.isi.edu/nsnam/ns>), we need to emphasize that such dedicated network simulators can emulate (mobile) network behaviour at the protocol/signal level.

Observe that in the abstract MANET model of Crowcroft, Gibbens, Kelly and Östring (2004) user data transmission are modelled by assigning flows (a macroscopic fluid-level model), abstracting from simulation at packet level: Estimation of the relevant performance indicators like data rates, resource utilization and the ratio of successful to unsuccessful transmissions does not require such a detailed low-level model. UC and UF signals, though, are explicitly included in model since they are necessary to determine data rates (data rates may change on receipt), yet this does not imply that these control signals *themselves* have to be modelled bit by bit at the protocol level: only the *impact* caused by the receipt of the control signals (data rate adjustments, another control signal being sent after a delay, yielding price propagation subject to delays) has to be represented in the model.

In summary, we consider DESMO-J to be an adequate choice for simulating the abstract MANET model because the requirements as established above are met, while the low-level protocol details (which are the strength of dedicated telecommunication network simulators) are not necessary.

4. MANET SIMULATION WITH DESMO-J

This section demonstrates how the abstract MANET environment described in Section 3 is implemented in DESMO-J. In particular, the **Event** classes that determine the behaviour of the network are described in detail using UML activity diagrams. The section is completed by describing a sample experiment and discussing the results.

4.1. Simulator Development

A simulation implementation in DESMO-J that uses the event-oriented view requires at least three types of components:

A. Entities

Entities are the relevant objects that the system consists of and that determine the behaviour of the system have to be identified. Note that these entities do not necessarily persist permanently: they may be created and discarded at some instant in time. In the MANET model, the entities to be included are *mobile node*, *transmission*, *UC signal* and *UF signal*.

For each of these entity types, a subclass of **Entity** has to be derived; since both signal types share some attributes, like storing the route to traverse, they should be derived from a common superclass, which in turn is a subclass of **Entity**. Table 1

summarizes their most important attributes. Other entity types are not strictly necessary, though such design decisions typically leave a degree of freedom to the user, who might decide to encapsulate a route into an entity of its own instead of route attributes being represented as arrays of nodes in transmission and signal entities. Likewise, the credits possessed by the nodes and their power and bandwidth resources are assigned to nodes in terms of numerical attributes.

Table 1: Entities of the MANET model

<i>Entity type</i>	<i>Selected attributes</i>
Mobile node	Position; movement pattern; power, bandwidth available, in use, prices; current transmissions; credit balance
Transmission	Route; data rate; data already sent, left to transmit
UC signal	Route; costs gathered
UF signal	Route; data rate; credits transferred

B. Events

Once the entities are defined, the model behaviour is described in terms of subclasses of **Event**, representing events that refer to a certain entity (see Section 2.2) such as a particular transmission terminating, and in terms of subclasses of **External Event**, representing events not referring to a certain existing entity, like a new node joining the network. Six of the eight types of events listed below are described in some detail using the UML activity diagram notation in Fig. 3, subject to two stereotypes: **<<create>>** refers to the creation of an entity, while **<<schedule>>** represents operations that modify the simulator's event list (scheduling, rescheduling or canceling an event); see Knaak and Page (2006) for more details about the usage of UML activity diagrams for discrete event simulation modelling. The two events not shown in Fig. 3, *update positions* and *update statistics* are sufficiently simple that the UML activity diagrams are omitted.

- *Transmission starts* (TS, Fig. 3a, **Event**, refers to the node of origin) initiates a new data transmission. If a route exists, the transmission is initiated, which requires sending an UF signal, see Fig. 2. The next transmission is scheduled.
- *Transmission completed* (TC, Fig. 3b, **Event**, refers to a transmission) terminates a data transmission, which requires sending an UF signal.
- *Node arrival* (NA, Fig. 3c, **External Event**) creates a new node that joins the network. Its first data transmission is scheduled as well as the arrival of the next node.

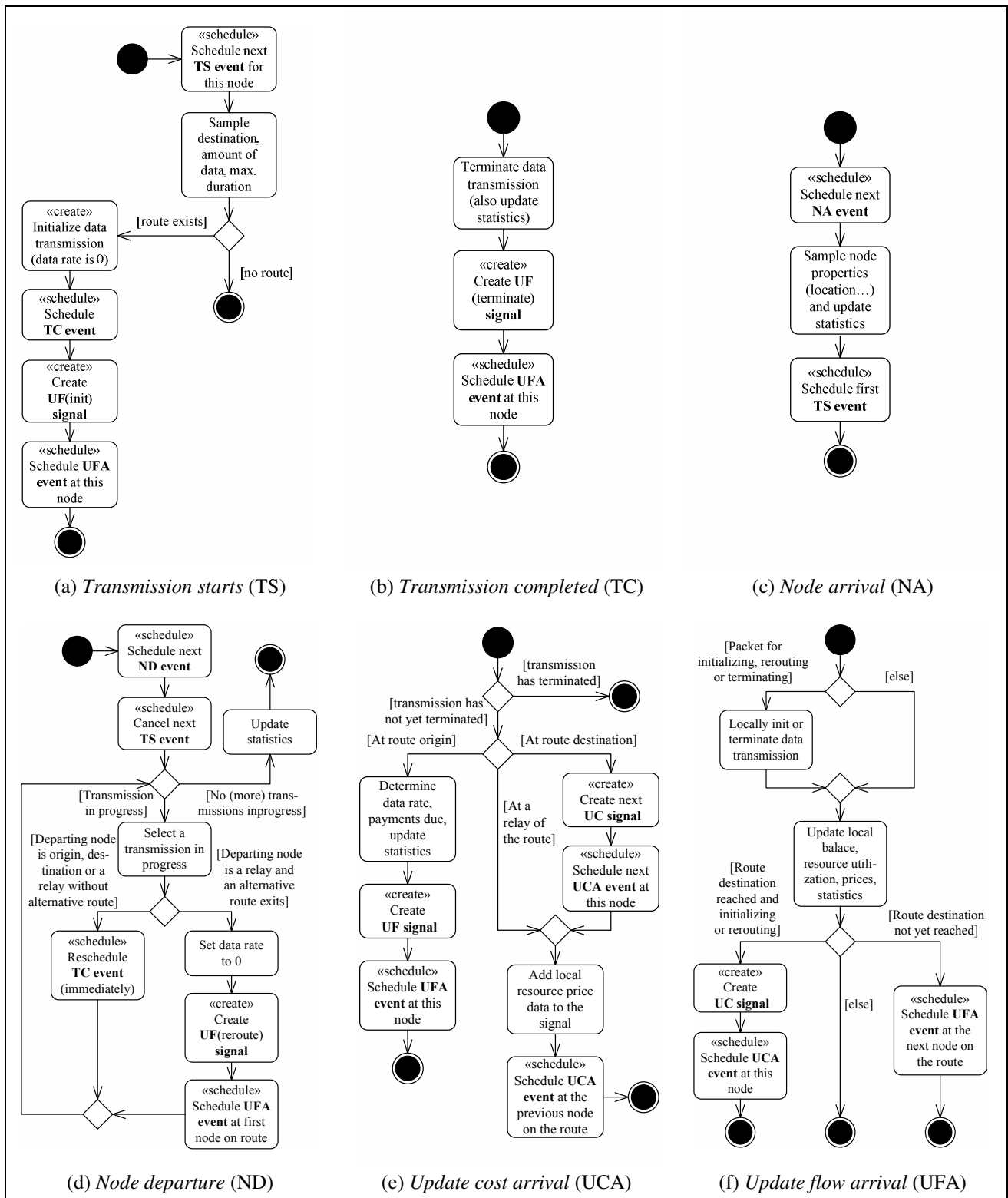


Figure 3: UML activity diagrams describing the behaviour of the MANET model (Knaak and Page 2006)

- *Node departure (ND, Fig. 3d, External Event)* removes a node from the network. Transmissions that are relayed through this node are rerouted (if possible) and transmissions that originate at this node are cancelled. The next node departure is scheduled.
- *Update cost arrival (UCA, Fig. 3e, Event, refers to an UC signal)* represents an UC signal arriving at a node. Provided the relevant transmission is neither completed nor cancelled, local cost data are collected and forwarded. The destination schedules the next UC signal. If the UC signal arrived at the

origin, the data rate is adjusted and an UF signal is sent.

- *Update flow arrival* (UFA, Fig. 3f, **Event**, refers to an UF signal) denotes an UF signal arriving at a node. The local credit balance is updated, the origin is debited and the relays and the receiver are credited. In addition, credit discounting (Section 3.2) is applied. Further signals (forwarding this UF signal towards the destination or issuing an UC signal at the destination in reply to an initializing/rerouting UF signal) are sent if applicable.
- *Update positions* (UP, not shown, **External Event**) updates the positions of the nodes according to their movement patterns and schedules the next UP event.
- *Update statistics* (US, not shown, **External Event**) updates various statistics and schedules the next US event.

C. Model

A class derived from **Model** is required to setup the parameters and an initial configuration of the nodes (as parsed from the XML experiment scenario file) and to trigger the dynamic behaviour of the model in terms of the events being scheduled, namely the first transmission starting (TS) on each node. The events responsible for the arrival and departure of the nodes (if desired) and updating the positions and the statistics are entered into the event list.

4.2. Experiment Configuration

For a sample experiment, consider a 12×12 Manhattan grid on a $320\text{m} \times 320\text{m}$ plane. Mobile nodes, which may be interpreted as pedestrians carrying mobile networking devices, traverse this grid, proceeding from one randomly selected target (street intersection) to the next, subject to varying speeds and to varying delays once a target is reached or (for a shorter period) before other intersections are traversed.

Once per second (exponentially distributed), each mobile node ($\Gamma_j = 0.5\text{W}$, $C_j = 10\text{Mb}$) initiates a data transmission of on average 2Mb (exponentially distributed). The target is either another mobile node (75%) or an external agency which is referred to as the outside world (25%). Nodes also may receive data transmissions from the outside world. Data transmissions to and from the outside world are routed via access points (APs) which is part of a fixed infrastructure (FI). The APs are connected to the outside world and the APs are logically fully meshed, so that a wired (AP to AP) hop can be used for data transmissions from one mobile node to another. Whether or not to include the FI in mobile to mobile

routes and which AP to use for a data transmission to/from the outside world is determined by least-cost routing considerations. Note that unlike the mobile nodes and the outside world, the APs do not originate data transmissions.

The provider of the FI has to determine the number and the spatial distribution of the APs and how much transmission power and bandwidth to allocate to them. Fig. 4 shows that the APs may for example be located at the centres of hexagonal cells whose radius is equal to the transmission range of the mobile nodes, which is assumed to be 50m . This hexagonal pattern ensures that every mobile node is always in radio contact with at least one access point: it can be proved that no pattern exists that provides this property using fewer access points (Lim and Lee 1997).

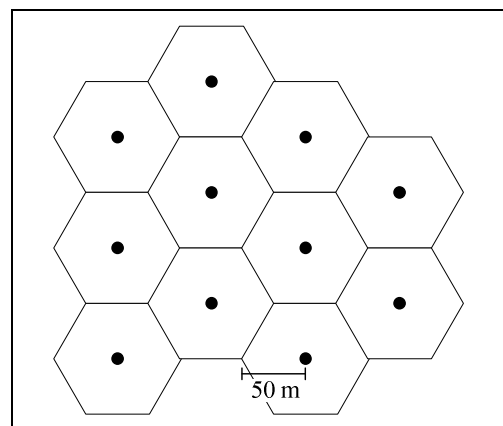


Figure 4: Hexagonal cells, radius 50 m

Assume the FI provides considers two alternative FI configurations: The *dense* FI configuration uses hexagonal cells of radius 50m to ensure that every mobile node is always in direct radio contact with at least one AP. This requires 15 APs. Initial simulation studies show that an AP resource configuration of $\Gamma_j = 5\text{W}$, $C_j = 50\text{Mb}$ is sufficient to carry all data transmissions. The *sparse* FI configuration allocates (approximately) the same power and bandwidth resources at only 8 APs ($\Gamma_j = 10\text{W}$, $C_j = 100\text{Mb}$). The resulting hexagonal cells of radius 65m do not guarantee that every mobile node is always in *direct* radio contact with at least one AP.

4.3. Results

Simulation runs using both the dense and the sparse FI configurations described above were executed 10 times for 10,000 seconds each, so that each node attempts approximately 10,000 transmissions, using different random number generator seeds. Table 2 shows some important performance indicators and their 95%

Table 2: Experiment Results

<i>fixed infrastructure</i>	<i>avg. FI resource usage</i>		<i>avg. transmission throughput (Mb/s)</i>			<i>transmissions completed</i>
	<i>power (W)</i>	<i>bandwidth (Mb)</i>	<i>via FI</i>	<i>not via FI</i>	<i>total</i>	
dense	21.2 ± 0.2	593.8 ± 6.9	195.4 ± 1.7	53.6 ± 1.6	249.0 ± 0.1	$98.2\% \pm 0.0\%$
sparse	24.3 ± 0.1	584.3 ± 2.3	190.5 ± 0.6	47.6 ± 0.4	238.1 ± 0.5	$92.7\% \pm 0.3\%$

confidence intervals.

We observe that in both FI configurations, the power and bandwidth of the APs namely $\sum \Gamma_j \geq 75 \text{ W}$, $\sum C_j \geq 750 \text{ Mb}$ are not fully utilized. This indicates that the resources provided by the service provider are sufficient, so that the traffic demand and the transmission capabilities of the mobile nodes are the bottlenecks that prevent the nodes from completing all their data transmissions successfully. The dense configuration achieves a slightly larger throughput for the calls routed via the FI than in the sparse configuration: this corresponds to more bandwidth in use. Nevertheless, the power used is smaller: on average, the mobile nodes are located closer to the APs, which decreases the transmission energy required.

Note that the dense FI configuration also increases the throughput of the data transmission that are *not* routed via the FI. This is because, in comparison to the sparse FI configuration, less transmission capacity is required at the mobile nodes (which are bottlenecks) to relay FI-based transmissions to and from the APs. Both the transmissions routed via the FI and the transmissions not routed via the FI benefit from this additional capacity.

Which FI configuration should the service provider adopt? This question cannot be answered without further economic details – yet it seems unlikely that the marginal increase in data throughput and the savings in energy consumption, which both translate to increased revenue, compensate for the cost of deploying and maintaining almost twice as many APs. This leads to a recommendation of the sparse FI configuration.

5. CONCLUSIONS

This paper has introduced DESMO-J, a framework for discrete event simulation in Java, whose key features include flexibility and joint event- and process-oriented model implementation. As an example of a simulation application besides the traditional areas of logistics such as queue-based production systems and transportation, a MANET simulation is used to present how to design and implement a simulator based on DESMO-J. The essential features of the MANET model, including the underlying economics, the decentralized protocol of data rate allocation and a relatively simple simulation experiment. Further information and more comprehensive experiments can be found in Göbel and Krzesinski (2008b).

DESMO-J is continuously being improved and extended; refer to <http://www.desmoj.de> for the latest updates. An environment for simulation conduction based on the *Eclipse* (<http://www.eclipse.org>) platform is scheduled for release in autumn 2009; see Czogalla, Knaak and Page (2006) for an early outline.

REFERENCES

- Camp, T., Boleng, J. and Davies, V., 2002. A survey of mobility models for ad hoc network research. *Wireless Communications and Mobile Computing* 2:483-582.
- Crowcroft, J., Gibbens, R., Kelly, F. and Östring, S., 2004. Modelling Incentives for Collaboration in Mobile Ad Hoc Networks. *Performance Evaluation* 57(4):427-439.
- Czogalla, R., Knaak, N. and Page, B., 2006. Simulating the Eclipse Way: A Generic Experimentation Environment based on the Eclipse Platform. *Proceedings of the 20th European Conference on Modelling and Simulation (ECMS) 2006*, pp. 260-265. May 28-31, Bonn (Germany).
- Göbel, J. and Krzesinski, A. E., 2008a. A Model of Autonomous Motion in Ad Hoc Networks to Maximise Area Coverage. *Proceedings of the Australasian Telecommunication Networks and Applications Conference (ATNAC) 2008*. Dec 7-10, Adelaide (Australia).
- Göbel, J. and Krzesinski, A. E., 2008b. Modelling Incentives and Protocols for Collaboration in Mobile Ad Hoc Networks. *Proceedings of the 11th ACM International Conference on Modeling, Analysis and Simulation of Wireless and Mobile Systems (MSWiM) 2008*. Oct 27-31, Vancouver (Canada).
- Göbel, J., Krzesinski, A. E. and Mandjes, M., 2007. Analysis of an Ad Hoc Network with Autonomously Moving Nodes. *Proceedings of the Australasian Telecommunication Networks and Applications Conference (ATNAC) 2007*, pp. 41-46. Dec 2-5, Christchurch (New Zealand).
- Knaak, N., Meyer, R. and Page, B., 2004. Logistic Strategies for Sustainable City Courier Services – An Agent-based Simulation Approach. *Proceedings of the International Workshop on Harbour, Maritime & Multimodal Logistics Modelling and Simulation (HMS) 2004*, pp. 139-144. Sep 16-18, Rio de Janeiro (Brazil).
- Knaak, N. and Page, B., 2006. Applications and Extensions of the Unified Modeling Language UML 2 for discrete Event Simulation. *International Journal of Simulation* 7(6):33-43.
- Lim, K. and Lee, Y.-H., 1997. Optimal partitioning of heterogeneous traffic sources in mobile communications networks. *IEEE Transactions on Computers* 46(3):312-325.
- Page, B. and Kreutzer, W., 2005. *The Java Simulation Handbook – Simulating Discrete Event Systems with UML and Java*. Aachen (Germany): Shaker.
- Page, B. and Kreutzer, W., 2006. A Framework for Web-based E-Learning of Discrete Event Simulation Concepts. *Proceedings of the Simulation und Visualisierung 2006*, pp. 195-204. Mar 2-3, Magdeburg (Germany).
- Page, B. and Neufeld, E., 2003. Extending an Object-oriented Discrete Event Simulation Framework in Java for Harbour Logistics. *Proceedings of the*

International Workshop on Harbour, Maritime & Multimoda Logistics Modelling and Simulation (HMS) 2003. Sep 18-20, Riga (Latvia).

Reddy, T. B., Karthigeyan, I., Manoj, B. and Murthy, C. S. R., 2006. Quality-of-service provisioning in ad hoc wireless networks: a survey of issues and solutions. *Journal of Ad Hoc Networks* 4(1):82-124.

Wittmann, J., Göbel, J., Möller, D. and Schroer, B., 2007. Refinement of the Virtual Intermodal Transportation System (VITS) and Adoption for Metropolitan Area Traffic Simulation. *Proceedings of the 2007 Summer Computer Simulation Conference (SCSC'07)*, pp. 587-592. Jul 15-18, San Diego (California, USA).

Wohlgemuth, V., Page, B., Mäusbacher, M. and Staudt-Fischbach, P., 2004. Component-Based Integration of Discrete Event Simulation and Material Flow Analysis for Industrial Environmental Protection: A Case Study in Wafer Production. *Proceedings of the 18th International Conference for Environmental Protection*, pp. 303-312. Oct 21-23, Geneva (Switzerland).

AUTHORS BIOGRAPHY

J. Göbel holds a diploma in Information Systems from the University of Hamburg, Germany. He now works as scientific assistant and PhD candidate at the Center of Architecture and Design of IT-Systems at the University of Hamburg; research interests focus on discrete event simulation and self-organizing transport networks.

A.E. Krzesinski obtained the MSc from the University of Cape Town and the PhD from Cambridge University, England. He is a Professor of Computer Science at the University of Stellenbosch, South Africa. His research interests centre on the performance evaluation of communication networks.

B. Page holds degrees in Applied Computer Science from the Technical University of Berlin, Germany, and from Stanford University, USA. As professor for Applied Computer Science at the University of Hamburg he researches and teaches in the field of Computer Simulation as well as in Environmental Informatics.

A CONSTRAINT PROGRAMMING-BASED LIBRARY FOR THE VEHICLE ROUTING PROBLEM

Daniel Riera^(a), Àngel A. Juan^(b), Daniel Guimarans^(c) Estel·la Pagans^(d)

^{(a)(b)}GRES-UOC/DPCS. EIMT. Universitat Oberta de Catalunya

^(c) Telecommunication and System Engineering Dept., Universitat Autònoma de Barcelona, Spain

^(d)Odournet S.L., Barcelona, Spain

^(a)drierat@uoc.edu, ^(b)ajuanp@uoc.edu, ^(c)daniel.guimarans@uab.cat, ^(d)epagans@uoc.edu

ABSTRACT

This paper presents a first approach to a constraint programming-based library for the modelling of general vehicle routing problems (VRP). In this work we present a set of constraints and cost functions written in ECLiPSe, a well known Constraint Programming (CP) language, which can be combined to build different VRP models without rewriting.

Keywords: vehicle routing problem, constraint programming, library

1. INTRODUCTION

Libraries are a nice facility when a researcher is trying to solve a problem and wants to improve something somebody else has proposed before. Although previous results are usually published and easily accessible, often it is not easy to reach these results the same way the original author did. This means a lot of time and perhaps a few e-mails to be able to exactly repeat something already existing.

Our research groups (i.e. GRES-UOC and DPCS) work on hybrid approaches to solve NP problems. Currently, the chosen one is the VRP (Golden, Raghavan and Wasil 2008). The VRP is a well known NP-Hard problem consisting in the service of a set of clients - spread in a map - by a set of vehicles. This is the generalisation of multiple different specific problems, which differ in the definition of the constraints applied to the basis. Thus, a few examples are: the CVRP, which contain constraints on the capacity of the vehicles; the VRPTW which includes time windows when clients can be served; etc.

Although there are specific approaches for these complex problems, single techniques often fall short to tackle real world instances. These can be easily seen because results are available only for laboratory problems, or because models become a simplification of the real problem. The later usually makes too big the distance between the solution and the original problem, and thus it becomes useless.

A common solution to the problems presented in the previous paragraph is the hybridization. It implies the use of different techniques/methodologies like

artificial intelligence, operations research or constraint programming in order to cooperate and find better ways to reach a solution. These techniques have their advantages and disadvantages, but it is very important for us to be able to build a model in any of their formalisms as quick as possible. A common mistake is the adaptation of models built for a specific methodology - like simplex, for instance - to another one - like constraint programming. Since the solving algorithms, methods, etc. are different, the model written thinking in one of these technologies is terribly slow and inefficient when used in the other.

Researchers are usually experts in several methodologies but they do not need to be in all of them. The use of a library thought for a specific programming methodology - in this case, constraint programming - helps them and allows researchers to think about the problem itself and avoid the modelling issues.

This paper presents a first approach to develop a library for modelling the VRP by means of CP. The main aim of the paper is to offer a tool not only for being used by our research groups but also by the research community working around the VRP. The use of this library leaves the constraint programming model implementation hidden from the user. Thus, programming issues are leaved apart and model construction becomes straightforward.

The structure of the paper is as follows. After this introduction, section 2 talks about the problem studied (the vehicle routing problem) and the different instances which can be found in literature. Section 3 gives an overview of the techniques commonly used to tackle the VRP. Later, sections 4 and 5 present the mathematical model - as a set of variables and constraints on those variables - used for the library construction and the library Prolog/ECLiPSe (Sterling and Shapiro 1986) predicates corresponding to those constraints. While section 6 explains how to create a file with a VRP model, section 7 shows the results of a case study. Finally, the conclusions and future work conclude the paper in section 8.

2. VEHICLE ROUTING PROBLEMS

The Vehicle Routing Problem (VRP), as told before, is an NP-hard problem in which a set of customers' demands have to be served by a fleet of vehicles departing from a certain number of depots. This is the basis to a very heterogeneous family of problems which can include extra constraints, enriching the possible model of the problem to solve. Furthermore, there are some costs associated with the resolution of these problems. In particular, it is usual to explicitly consider costs due to moving a vehicle from one node - customer or depot - to another. The classical goal here consists on determining the optimal set of routes that minimizes those tangible costs under a certain set of predefined constraints.

Depending on the considered constraints there are a number of VRP versions, but firstly, there is an initial classification to consider: On the one hand, there are the static VRPs. These are all those problems with no additional elements to be incorporated in real-time. Thus, all the information is known from the beginning. They are clearly static and deterministic. On the other hand there are problems which consider that customers, constraints or others might be added later, and hence, a re-routing would be needed on-line. These are called dynamic VRPs. Although usually, the main difference between these two versions should not have too much to do with the problem model and the optimisation strategy, when working with stochastic demands the model might need some adaptation. In the current work stage, dynamic VRPs with stochastic demands are not considered but everything said further can be applied to the rest of static and dynamic VRPs.

These are some of the most studied VRPs:

- CVRP (Capacitated VRP): Vehicles have limited carrying capacities
- VRPTW (VRP with Time Windows): Clients' locations have time windows within they must be served
- VRPPD (VRP with Pick Up and Delivery): Goods need to be moved from certain pickup clients to other delivery clients
- VRP LIFO (VRP with LIFO): Similar to the VRPPD, except an additional restriction is placed on the loading of the vehicles: at any delivery client, the item being delivered must be the item most recently picked up
- MDVRP (Multiple Depots VRP): Vehicles start from two or more depots
- PVRP (Period VRP): The VRP has to be solved for two or more days (in contrast to normal VRPs which work with one day routing)
- SDVRP (Split Delivery VRP): It is allowed that the same customer can be served by different vehicles
- VRPB (VRP with Backhauls): Customers can demand or return some commodities

- VRPSF (VRP with Satellite Facilities): Includes de use of satellite facilities to replenish vehicles during a route
- VRPSD (VRP with Stochastic Demands): Includes probabilistic descriptions of the demands to be served.

Notice that these are only a few pure VRPs but combinations of these could also appear in literature. Furthermore, the wide variety of cost functions adds more possibilities when a new VRP is modelled. Costs may include:

- Routes distances: These are clearly related both to the routing total time and the oil used to complete it
- Resources needed: The number and kinds of vehicles are a key issue, not only due to their direct costs but also for the human resources (drivers) needed
- Environmental issues: Transport is an important variable in environmental care not only for the atmospheric harm or the traffic jams but also due to noises, odours, etc.
- Intangible costs: Competence among enterprises, workers happiness or visual quality of solutions given, often are not considered but very important in decisions taking

3. RELATED WORK

The VRP has been studied for long and from several points of view. That is, different problem versions (see previous section) and different methodologies (e.g. artificial intelligence, operations research, simulation, etc.) have been the aim of many researchers work. A recent survey of the VRP can be found at Sterling and Shapiro (1986).

Both the VRP and VRPTW problems have been formulated as an Integer Linear Programming (ILP) problem. A recent survey on the proposed formulations based on ILP is Letchford and Salazar-González (2006). Other complete formulations are those based on Column Generation (e.g. Desaulniers, Desrosiers and Solomon (2005)) or Lagrangian Relaxation (e.g. Fisher and Jörnsten (1997)).

Since VRP is an NP-Hard problem, and hence, optimality prove becomes impossible. Therefore, several heuristic solution methods have been used. Construction methods like, for instance, the "Savings" method of Clarke and Wright (1964) have proved to work very well. There are other approaches to routes construction, like those including the "sweep" heuristic of Gillet and Miller (1974).

Local search methods have been deeply used too. These algorithms try to improve the current solution by moving to its neighbours depending on specifically defined "move operators". Two examples of these operators, commonly used together with constraint programming are Guided Local Search (Shaw 1998)

and Limited Discrepancy Search (Harvey and Ginsberg 1995).

Metaheuristics go a step further, trying to avoid local minima. Many metaheuristics have been successfully applied to the VRP. Perhaps, the most important ones are Simulated Annealing (Bent and Van Hentenryck 2004), Tabu Search (Cordeau, Laporte and Mécier 2001), or Genetic Algorithms (Homerger and Gehring 2005).

Given the complexity of real-world VRP problems, Constraint Programming complete search would be too inefficient to be considered by itself. Anyway, its combination with other local search methods can take advantage of a CP framework. The library presented in this paper allows experts on other techniques to combine their solutions with models written specifically for CP. Notice that all these methodologies often fall short to work with real-life VRP problems. Moreover, every technique has got its own drawbacks or weaknesses. Hybrid approaches seem to overcome single-methodology results (e.g. Juan, Faulin, Riera, Masip and Jorba (2009)).

4. THE VRP MODEL

4.1. Notation

For the model constraints and costs defined, the following notation has been used (notice that later, in the constraints definition, sometimes they may appear in lower case):

- $C = C_1 \dots C_n$ are the customers to serve.
- $M = M_1 \dots M_m$ are the available vehicles.
- $Qm = Qm_1 \dots Qm_m$ are the vehicles capacities.
- $V = V_1 \dots V_{n+2m}$ are the visits, with domain $1 \dots m$.
- Two sublists of V , F and L , are defined as the vehicles departure and arrival nodes.
- For each visit V_i there exist the predecessor of that visit, P_i , and its successor, S_i .
- R_i is the amount of goods to pick up in visit i .
- After every visit, Q_i is the quantity of goods in the vehicle serving the visit.
- Visit i is performed in time T_i .
- In terms of costs, τ_{ij} and δ_{ij} represent the travel time and the distance from visit i to visit j , and τ_i is the time spent for pick up or delivery in i .

4.2. Constraints

The first approach to the library contains the following constraints:

4.2.1. Difference constraints

Equations (1) and (2) force predecessors and successors' lists to contain no repetitions. Thus, one client can have one and only one predecessor and successor.

$$\forall i, j \in V, i < j: p_i \neq p_j \quad (1)$$

$$\forall i, j \in V, i < j: s_i \neq s_j \quad (2)$$

4.2.2. Coherence constraints

Equations (3) and (4) connect the concepts successor and predecessor as follows: The former says that i is the successor of its predecessor. And the later says that i is the predecessor of its successor

$$\forall i \in V - F: s_{p_i} = i \quad (3)$$

$$\forall i \in V - F: p_{s_i} = i \quad (4)$$

4.2.3. Path constraints

Equations (5) and (6) are used to assure a route is visited by a single vehicle. Thus the vehicle assigned to i must be the same as those assigned to its predecessor and successor.

$$\forall i \in V - F: v_i = v_{p_i} \quad (5)$$

$$\forall i \in V - L: v_i = v_{s_i} \quad (6)$$

4.2.4. Capacity constraints

Equations (7) and (8) count the goods picked up in a route, and the goods delivered in that route. Thus, the first constraint says the goods accumulated after visiting client i is the addition of those accumulated in its predecessor plus those taken in i . The second is similar but using the successor.

$$q_i \geq 0$$

$$r_i \neq 0$$

$$\forall i \in V - F: q_i = q_{p_i} + r_i \quad (7)$$

$$q_i = q_{s_i} - r_{s_i} \forall i \in V - L \quad (8)$$

Furthermore, the maximum capacity defined for a vehicle Qm_j must limit the accumulated capacities for every client visited by that vehicle. This can be done either with an extra constraint or with domains bounding.

4.2.5. Time constraints

Equations (9) and (10) bound the accumulated time spent by a vehicle visiting client i . This time is at least, the accumulated time in the predecessor of i , plus the time spent in i . Equally, this time must be, at most, the accumulated time in the successor of i , minus the time spent in i .

$$t_i \geq 0$$

$$t_i \geq t_{p_i} + \tau_{p_i, i} \forall i \in V - F \quad (9)$$

$$t_i \leq t_{s_i} - \tau_{i, s_i} \forall i \in V - L \quad (10)$$

4.2.6. Time constraints 2

Equations (11) and (12) are similar to equations (9) and (10) including travel times from and to the predecessor and the successor of i respectively.

$$t_i \geq 0$$

$$t_i \geq t_{p_i} + \tau_i + \tau_{p_i,i} \forall i \in V - F \quad (11)$$

$$t_i \leq t_{s_i} - \tau_i - \tau_{i,s_i} \forall i \in V - L \quad (12)$$

4.3. Cost functions

4.3.1. Distance travelled

Equations (13) and (14) model the cost as the additions of the travelled distances.

$$d = \sum_{i \in V - F} \delta_{p_i,i} \quad (13)$$

$$d = \sum_{i \in V - L} \delta_{i,s_i} \quad (14)$$

5. THE VRP LIBRARY

In order to build a model using this library, variables for P, S, V, C, Q and T must be defined. Furthermore, input lists or matrices defining costs are needed.

Once the variables and domains are declared, the following rules have been programmed in ECLiPSe and can be called to model the constraints of the problem:

5.1. Programmed constraints

5.1.1. Difference constraints

constraintsDifferent(+P,+S,+CN)

5.1.2. Coherence constraints

constraintsCoherence(+P,+S)

5.1.3. Path constraints

constraintsPath(+V,+P,+S,+MN)

5.1.4. Capacity constraints

constraintsCapacity(+Q,+R,+P,+S,+MN)

5.1.5. Time constraints

constraintsTime(+T,+Times,+P,+S,+MN)

5.1.6. Time constraints 2

constraintsTime2(+T,+Times,+P,+S,+MN)

5.2. Programmed cost functions

5.2.1. Distance travelled

costFunctionTotalDistanceMin(+T,+MN,-Cost)

Notice that other constraints to remove symmetries or specifications of the above constraints are already

programmed and documented in the library. Moreover, new constraints and costs are being continuously added.

6. EXAMPLE OF CVRP USING THE LIBRARY

As said before, the library is written in ECLiPSe (2009). In order to write a model, the library (stored in the same directory as the model) is loaded as shown in Fig. 1.

```
:-[ 'vrp_lib.ecl' ].
```

Figure 1: VRP Library Call

Moreover, the Constraint Satisfaction Problem (i.e. variables, domains and constraints) must be declared. Later the search is performed and the solution is shown. The set of necessary calls can be seen in Fig. 2.

```
solve(B, H1, H2) :-
    variables(VARS, VARSN),
    domains(VARS, VARSN),
    constraints(VARS, VARSN),
    optimise(VARS, VARSN, B, H1, H2),
    showSolution(VARS, B).
```

Figure 2: General Rule to Solve a CSP with the VRP Library

Notice that H1 and H2 are the heuristics to perform the search. These are: the order the variables are assigned, and the order in which the possible values of a variable are chosen, respectively.

The rule *variables* (see Fig. 3) contains the definition of the necessary variables. These are usually lists, since they store one variable for each vehicle, customer, etc.

```
variables([C,M,V,Q,T,Times,R,Qv,P,
S],[CN,MN]) :-
    readCustomers(C,CN,MN,R,Times),
    readVehicles(M,MN,Qv),
    VN is CN + 2 * MN,
    P_SN is CN + MN,
    length(V,VN),
    length(Q,VN),
    length(T,VN),
    length(P,P_SN),
    length(S,P_SN).
```

Figure 3: CSP's Variables Definition

Notice that rules *readCustomers* and *readVehicles* are used to read the problem instance to be solved.

The rule *domains* (see Fig. 4) defines the feasible values the variables of the CSP might take. Normally, the smaller the domains are, the faster the search will be.

```

domains([C,M,V,Q,T,_,_,Qv,P,S],[CN
,MN]):-
(foreach(Ci,C),for(I,1,CN) do
Ci #= I),
(foreach(Mi,M),for(J,1,MN) do
Mi #= J),
decompose(V,MN,_,_,V_FL,_,_),
V_FL::M,
max(Qv,Max),
Q::0..Max,
T::0..500,
length(P,P_SN),
P :: 1..P_SN,
S :: 1..P_SN.

```

Figure 4: CSP's Domains Definition

The rule *constraints* (see Fig. 5) is a set of calls to the library constraints which must be included in the model.

```

constraints([_,_M,V,Q,T,Times,R,Qv
,P,S],[CN,MN]):-
constraintsCiclicRoutes(V,MN),
constraintsDifferent(P,S,CN),
constraintsCoherence(P,S),
constraintsPath(V,P,S,MN),
constraintsCapacity(Q,R,P,S,MN),
constraintsMaxCapacity(Q,V,Qv,MN),
constraintsTime2(T,Times,P,S,MN),
constraintsSymmetriesVisitsSorted(
V,MN),
constraintsSymmetriesAllVehiclesUs
ed(P,S,CN,MN).

```

Figure 5: CSP's Constraints Definition

Finally, the rule *optimise* (see Fig. 6) contains the necessary calls to optimise the model.

```

optimise([C,_,V,Q,T,_,_,_,P,S],[_,
MN],B,H1,H2):-
flatten([P,S],VARSflat),
flatten([V,C,Q,T],VARS2flat),
costFunctionTotalDistanceMin(T,MN,
Cost),
bb_min((search(VARSflat,0,H1,H2,co
mplete,[backtrack(B)]),indomain(Co
st)),Cost,bb_options{strategy:cont
inue}),once(labeling(VARS2flat)).

```

Figure 6: CSP's Search Definition

In order to perform the search, only predecessors and successors are instantiated. The rest of values of the model become either instantiated or bounded depending on propagation.

Notice that new improvements in models specification and search might be included after

publishing this paper. The search presented is the basic one. Download the example files from the URL introduced in section 8 in order to take advantage of new advances and to avoid problems with new versions.

7. CASE STUDY

In order to test the library, the following CVRP has been programmed:

$$C = [C_1, C_2, C_3, C_4, C_5, C_6, C_7, C_8]$$

$$M = [M_1, M_2, M_3]$$

$$Qm = [220, 220, 220]$$

$$R = [69, 80, 87, 38, 54, 122, 74, 91]$$

While the depot is placed in the origin (0,0), the 8 clients are respectively in (-49.95, -95.30), (-70.61, -13.54), (-1.01, 76.81), (-5.94, 20.97), (9.74, 18.50), (-70.42, 10.00), (95.50, -90.35), (35.32, 59.99).

For this example, in terms of cost, only travel times will be considered. Let us suppose, for simplification, that the travel time is defined as the integer part of the Euclidean distance between two points:

$$\tau = \begin{pmatrix} 0 & 84 & 178 & 124 & 128 & 107 & 145 & 155 \\ 84 & 0 & 114 & 73 & 86 & 23 & 183 & 81 \\ 178 & 114 & 0 & 56 & 59 & 96 & 193 & 38 \\ 124 & 73 & 56 & 0 & 15 & 65 & 150 & 48 \\ 128 & 86 & 59 & 15 & 0 & 80 & 138 & 61 \\ 107 & 23 & 96 & 65 & 80 & 0 & 193 & 61 \\ 145 & 183 & 193 & 150 & 138 & 193 & 0 & 199 \\ 155 & 81 & 38 & 48 & 61 & 61 & 199 & 0 \end{pmatrix}$$

Given this CVRP, after generating the CSP (using the library presented) the output of the ECLIPSe application is shown in Fig. 7.

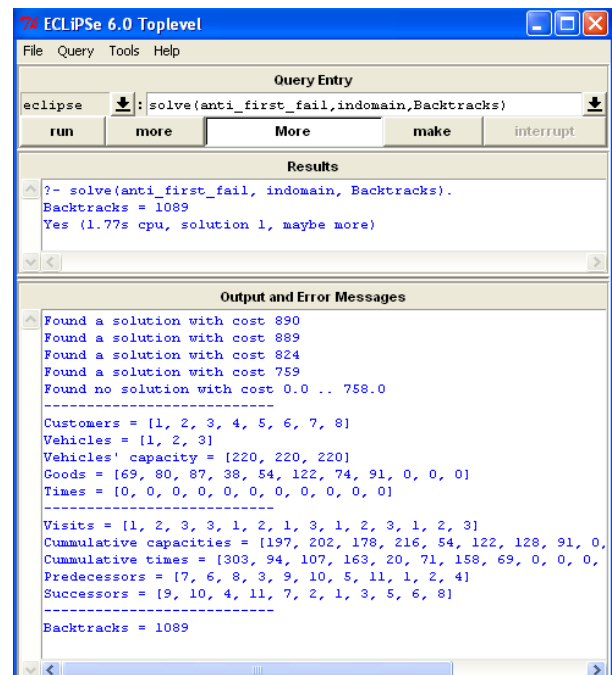


Figure 7: Result of CSP Model Optimisation

For better comprehension, the solution found is painted in Fig. 8.

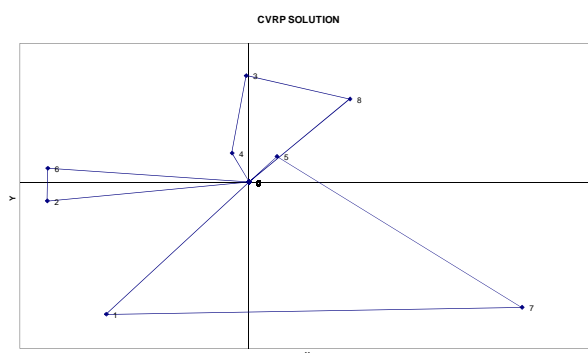


Figure 8: Optimal Routes Found

8. CONCLUSIONS AND FUTURE WORK

The aim of the presented work is to ease researchers and students working around the VRP their construction of models using constraint programming. The library is currently in an initial stage and only a few constraints and cost functions are available. On the one hand, we are working on adding first the most common constraints. On the other hand we are proposing new cost functions related to environmental and non tangible issues.

The idea is to include new constraints and costs to the library, completing it to allow the modelling of as much kinds of VRPs as possible. The library is available at the URL <http://gres.uoc.edu/VRP>. We expect to complete it thanks to comments and questions sent.

ACKNOWLEDGMENTS

The research of this paper has been partially sponsored by the Internet Interdisciplinary Institute (IN3) and the Knowledge Community on *Hybrid Algorithms for solving Real-life rOuting, Scheduling and Availability problems (HAROSA)*.

REFERENCES

- Bent, R., Van Hentenryck, P. A two-stage Hybrid Local Search for the Vehicle Routing Problem with Time Windows. *Transportation Science*, 38(4):515, 2004.
- Clarke, G. and J. Wright. Scheduling of Vehicles from a central Depot to a Number of DeliveringPoints. *Operations Research*, 12, 568-581, 1964.
- Cordeau, J.F., Laporte, G., Mécier, A. A Unified Tabu search Heuristic for Vehicle Routing Problems with Time Windows. *Journal of the Operations Research Society*, 52(8):928-936, 2001.
- Desaulniers, G., Desrosiers, J., Solomon, M.M. (editors) Column Generation. Springer, 2005.
- ECLiPSe Home. *The ECLiPSe Constraint Programming System*. Available from: <http://87.230.22.228/index.html> [accessed 2 June 2009]

- Fisher, M.L., Jörnsten, K.O. Vehicle Routing with Time Windows: Two Optimization Algorithms. *Operations Research*, 45(3):488-492, 1997.
- Gillet, B., Miller, L.R. A Heuristic Algorithm for the Vehicle Dispatch Problem. *Operations Research*, 22:340-349, 1974.
- Golden, B., Raghavan, S., Wasil, E. 2008. The vehicle routing problem. *Latest advances and new challenges*, Springer, New York (USA).
- Harvey, W.D., Ginsberg, M. L. Limited Discrepancy Search. *Proceedings of the Fourteenth International Joint conference on Artificial Intelligence (IJCAI-95)*, volume 1, pg. 607-615, Montreal (Canada), 1995.
- Homberger, J., Gehring, H. A two-phase Hybrid Metaheuristic for the Vehicle Routing Problem with Time Windows. *European Journal of Operation Research*, 162(1):220-238, 2005.
- Juan, A.A., Faulin, J., Riera, D., Masip, D., Jorba, J. A Simulation-based Methodology to assist Decision-makers in real Vehicle Routing Problems. *Proceedings of the 11th Int. Conf. on Enterprise Information Systems*. Milan, 2009.
- Letchford, A.N., Salazar-González, J.J. Projection results for vehicle routing. *Mathematical Programming*, 105(2):251-274, 2006.
- Rossi, F., Van Beek, P., Walsh, T. 2006. Handbook of Constraint Programming. *Foundations of Artificial Intelligence Series*, Elsevier, Amsterdam (The Netherlands).
- Shaw, P. Using Constraint Programming and Local Search Methods to Solve Vehicle Routing Problems. *Fourth International conference of Principles and Practice of Constraint Programming (CP'98)*, 1998.
- Sterling, L., Shapiro, E. 1986. The art of Prolog. *The MIT Press*. Cambridge (USA).
- Toth, P. and D. Vigo. 2002. The Vehicle Routing Problem. *SIAM Monographs on Discrete Mathematics and Applications*. SIAM
- Tsang, E., 1993. Foundations of Constraint Satisfaction. Academic Press.

A HYBRID CONSTRAINT PROGRAMMING / LOCAL SEARCH APPROACH TO THE PICK-UP AND DELIVERY PROBLEM WITH TIME WINDOWS

Daniel Guimarans^(a), Juan José Ramos^(b), Mark Wallace^(c), Daniel Riera^(d)

^{(a)(b)}Telecommunication and System Engineering Dept., Universitat Autònoma de Barcelona, Spain

^(c)Faculty of Information Technology, Monash University, Melbourne, Australia

^(d)Estudis d'Informàtica, Multimèdia i Telecomunicació, Universitat Oberta de Catalunya, Barcelona, Spain

^(a)Daniel.Guimarans@uab.cat, ^(b)JuanJose.Ramos@uab.cat, ^(c)Mark.Wallace@infotech.monash.edu.au, ^(d)drierat@uoc.edu

ABSTRACT

Routing vehicles to serve customers is a problem that naturally arises in many distribution systems. Moreover, fleet management requires fast algorithms able to cope with continuously changing needs. Many efforts have been addressed to tackle different vehicle routing problem's variants. Among them, the pick up and delivery problem with time windows (PDTW) has received far less attention despite its relevance from practical and theoretical perspectives. The present paper provides a hybrid approach to the PDTW based on Constraint Programming paradigm and local search. Indeed, the proposed algorithm includes some performance improvements to enhance its efficiency. Thus, this hybrid approach may provide a solution to problems otherwise intractable in a reasonable computational time, as shown in the presented results. Due to these characteristics, the proposed algorithm may be an efficient tool in decision making support, as well as a mechanism able to provide an initial solution for subsequent optimization techniques.

Keywords: pick up and delivery problem, hybrid algorithm, constraint programming, local search

1. INTRODUCTION

An important component of many distribution systems is routing vehicles to serve customers. In fact, many companies are faced with problems regarding the transportation of people, goods or information. In many cases, they have to efficiently manage a heterogeneous vehicle fleet providing pick-up and delivery services to a set of customers. These companies have to optimize transportation by using rational manners and effective tools.

This class of logistics problems is usually known as the vehicle routing problem (VRP) and its objective is usually twofold: to minimise travelling costs by arranging visits in a proper sequence, while reducing the number of vehicles to be used in order to serve all customers. In the field of combinatorial optimization, the VRP is regarded as one of the most challenging problems because of its NP-Hardness (Savelsbergh 1985). For such problems in real situations, it is often

desirable to obtain approximate solutions, so they can be found fast enough and are sufficiently accurate for the purpose.

The most basic VRP is the capacitated vehicle routing problem (CVRP) that assumes a fleet of vehicles of fixed capacity housed in a central depot. More realistic routing problems include travel times between every pair of nodes, customer service times and the maximum tour duration as additional problem data. These characteristics define the VRP with time windows (VRPTW) as a generalization of the CVRP. The VRP with pick-up and delivery and time windows, or pick-up and delivery problem with time windows (PDTW), is a generalization of the VRPTW. In the PDTW, pairs are defined among customers so pick up and drop off locations are determined, in addition to time windows constraints related to each visit.

Several classes of VRP have been studied in the literature. The VRPTW has been the subject of intensive research efforts for both heuristic and exact optimization approaches. Because of the high complexity level of the VRPTW and its wide applicability to real situations, solution techniques capable of producing good solutions in limited time, i.e. heuristics, are of major importance (Bräysy and Gendreau 2005a). Over last few years, many new heuristic approaches have been proposed, primarily metaheuristics, for tackling the VRPTW (Bräysy and Gendreau 2005b). With respect to the literature on VRPs, it is interesting to observe that although PDTW is as important and interesting from practical and theoretical points of view, it has received far less attention (Lau and Liang 2001; Savelsbergh and Sol 1995).

The present paper is structured as follow: section 2 introduces the proposed PDTW model, where the complete problem is decomposed into two separated subproblems. Next section includes a description of the adopted hybrid approach and algorithm's general structure, as well as some performance improvements added to the model. Section 4 presents some results obtained by using the proposed algorithm and corresponding analysis. Finally, conclusions and further research directions are outlined.

2. PROBLEM DESCRIPTION

The PDTW can be considered as a routing network, represented by a directed graph $G\{I, P, A\}$, connecting customer nodes $I = \{i_1, i_2, \dots, i_n\}$ and depot nodes $P = \{p_1, p_2, \dots, p_j\}$ through a set of directed edges $A = \{(i, j) \mid i, j \in (I \cup P)\}$. The edge $a_{ij} \in A$ is supposed to be the lowest cost route connecting node i to node j . At each customer location $i \in I$, a fixed load w_i is to be picked up or delivered by a single vehicle. A time window $[a_i, b_i]$ is defined for each customer i , where a_i is the earliest time and b_i is the latest time at which the service can start. Service time st_i at node i is also given.

A fleet of heterogeneous vehicles $V = \{v_1, v_2, \dots, v_m\}$ with different capacities q_v , located in a depot $p \in P$ is available to accomplish the required pick-up/delivery tasks. Each vehicle v must leave from the depot, perform corresponding pick-up and delivery orders without exceeding its capacity q_v at any time and then return to the same depot. In the present model, only one depot is considered, but the model can be easily extended to include multiple depots.

The route for vehicle v is a tour of nodes $r_v = (p, \dots, i_j, i_{j+1}, \dots, p)$ connected by directed edges belonging to A that starts and ends at depot p . Associated to the set of edges $a_{ij} \in A$, there is a pair of matrices $C = \{c_{ij}\}$ and $T = \{t_{ij}\}$ denoting the travel cost and the travel time from node i to node j , respectively. It is assumed that distances and times defined in the problem satisfy the triangular inequality, i.e. $c_{ik} + c_{kj} \geq c_{ij}$ and $t_{ik} + t_{kj} \geq t_{ij} \forall i, j, k \in (I \cup P)$.

A set of work orders $O = \{(i, j) \mid i \neq j; i, j \in (I \cup P)\}$ is defined among customers ($O \subseteq (I \cup P) \times (I \cup P)$) to relate pick-up and delivery locations. Thus, if vehicle v is assigned to serve the order (i, j) , denoted as o_{ij} , it must visit customer i to pick up the corresponding load w_i before visiting customer j , where a quantity w_j is to be delivered. In any case, both customers i and j related through an order o_{ij} should be assigned to the same vehicle's v route (5).

The proposed mathematical formulation, based on the MILP formulation presented in (Dondo and Cerdá 2007), requires defining two different sets of binary variables:

- The allocation variable Y_{iv} to assign vehicle $v \in V$ to customer $i \in I$, taking value 1. Otherwise, it is 0.
- The precedence variable S_{ij} to denote that customer $i \in I$ is visited before ($S_{ij} = 1$) or after ($S_{ij} = 0$) customer $j \in I$. It should be noted that this approach uses the notion of generalised predecessor rather than direct predecessor.

In the present model, PDTW has been divided into two subproblems, concerning resource's allocation and vehicle's routing separately. Allocation variable Y_{iv} is determined by solving the allocation subproblem, while precedence variable S_{ij} is set on the routing part. Furthermore, in the routing subproblem, values

obtained for Y_{iv} variables are used. Thus, the allocation variable Y_{iv} is shared between both subproblems.

Allocation and routing subproblems have been formulated separately according to the formalisms to be used. Thus, Constraint Programming (CP) paradigm has been used to partially define the allocation subproblem, while routing calculation has been tackled from a MILP perspective.

The proposed allocation subproblem has been formulated as follow:

$$\min \sum_{v \in V} cf_v B_v \quad (1)$$

s.t.

$$occurrences(1, Y_{iv}, 1) \quad \forall i \in I \mid i > 1 \quad (2)$$

$$\sum_{i \in I} w_i Y_{iv} = 0 \quad \forall v \in V \quad (3)$$

$$w_i Y_{iv} \leq q_v \quad \forall i \in I, v \in V \quad (4)$$

$$Y_{iv} = Y_{jv} \quad \forall o_{ij} \in O, v \in V \quad (5)$$

where cf_v is the fixed cost associated to using vehicle v .

The objective function (1) aims to minimise the number of vehicles used in the solution. A boolean variable B_v has been defined for each vehicle $v \in V$ to determine whether v is used in the solution ($B_v = 1$) or not ($B_v = 0$), according to the expression

$$B_v n \geq \sum_{i \in I} Y_{iv} \quad (6)$$

being n the total number of customers.

Constraint (2) states that every customer node $i \in I$ must be serviced by a single vehicle $v \in V$. Equation (2) is a CP global constraint equivalent to the linear expression

$$\sum_{v \in V} Y_{iv} = 1 \quad \forall i \in I \mid i > 1 \quad (7)$$

Capacity constraint (3) states that the overall load along the route of a vehicle $v \in V$ should be zero. Thus, all vehicles should be empty when departing or finishing their respective routes.

Expression (4) ensures that vehicle v assigned to serve customer i has enough capacity to pick up the corresponding load w_i . This constraint is meaningless for delivery locations, since their demand is always non-positive.

The MILP formulation for the routing subproblem is presented next:

$$\min \sum_{v \in V} (CV_v + c_i TV_v + \rho_v \Delta T_v) + \sum_{i \in I} \rho_i (\Delta a_i + \Delta b_i) \quad (8)$$

s.t.

$$C_i \geq c_{pi} Y_{iv} \quad \forall i \in I, v \in V \quad (9)$$

$$C_j \geq C_i + c_{ij} - M_c(1 - S_{ij}) - M_c(2 - Y_{iv} - Y_{jv}) \quad (10a)$$

$$C_i \geq C_j + c_{ji} - M_c S_{ij} - M_c(2 - Y_{iv} - Y_{jv}) \quad (10b)$$

$$\forall v \in V; \forall i, j \in I \mid i < j$$

$$CV_v \geq C_i + c_{ip} - M_c(1 - Y_{iv}) \quad \forall i \in I, v \in V \quad (11)$$

$$T_i \geq t_{pi} Y_{iv} \quad \forall i \in I, v \in V \quad (12)$$

$$T_j \geq T_i + st_i + t_{ij} - M_t(1 - S_{ij}) - M_t(2 - Y_{iv} - Y_{jv}) \quad (13a)$$

$$T_i \geq T_j + st_j + t_{ji} - M_t S_{ij} - M_t(2 - Y_{iv} - Y_{jv}) \quad (13b)$$

$$\forall v \in V; \forall i, j \in I \mid i < j$$

$$TV_v \geq T_i + st_i + t_{ip} - M_t(1 - Y_{iv}) \quad \forall i \in I, v \in V \quad (14)$$

$$\Delta a_i \geq a_i - T_i \quad \forall i \in I \quad (15)$$

$$\Delta b_i \geq T_i - b_i \quad \forall i \in I \quad (16)$$

$$\Delta T_v \geq TV_v - tv_v^{\max} \quad \forall v \in V \quad (17)$$

$$Q_i \leq q_v + M_{pd}(1 - Y_{iv}) \quad \forall i \in I, v \in V \quad (18)$$

$$Q_i \geq w_i + \sum_{j=i; j \in I} w_j B_{ij} \quad \forall i \in I \quad (19)$$

where c_t is the cost per unit time; CV_v and TV_v are the total distance and time for vehicle v , respectively; C_i and T_i are the cost and travel time from the depot to node i , respectively; Q_i is the load of vehicle v at customer i ; Δa_i and Δb_i are time window violations at customer i ; ΔT_v is the time window violation for vehicle v ; B_{ij} is a boolean variable; M_c , M_t and M_q are large positive numbers.

Problem's objective function (8) aims to minimise the overall service expenses, including travelling distance and time costs. Moreover, time window violations are included in the last two terms, penalizing non-fulfilment on either the maximum allowed working time (ΔT_v) or customers' time windows (Δa_i and Δb_i).

Equation (9) states that the cost of travelling from depot p to customer i (C_i) should be greater than or equal to the least travel cost from the depot to node i (c_{pi}). This constraint becomes binding iff customer i is the first visit included in vehicle's v route.

Constraint (10a) states that the distance based travel cost from the depot to customer j (C_j) should be greater than C_i by at least c_{ij} whenever customers i and j are included in the same route ($Y_{iv} = Y_{jv} = 1$, for some vehicle v) and i is visited first ($S_{ij} = 1$). In case customer j is visited earlier ($S_{ij} = 0$), the reverse statement (10b) holds. Constraints (10a) and (10b) both become redundant if nodes i and j are serviced by different vehicles. Furthermore, C_j values are determined only by previous visit's cost, due to triangular inequality ensures that C_i values are monotonically increasing. In addition, the latter prevents from cycles appearing in a feasible solution without adding specific constraints.

Expression (11) states that the overall travelling cost incurred by vehicle v (CV_v) must always be greater than, or equal to, the travelling expenses from the depot to any customer i (C_i) by at least the amount c_{ip} . Indeed, triangular inequality guarantees that the last node visited by vehicle v is the one finally binding the value of CV_v .

Constraints (12)-(14) are time equivalents to distance based cost constraints (9)-(11). Thus, properties described above still hold since expressions

are only modified by adding the service time at customer i (st_i).

Time windows can be hard (HTW) or soft (STW). When the time windows are regarded as hard constraints, expression (15) states that a vehicle cannot start the service at the assigned customer i before the earliest time a_i by simply making $\Delta a_i = 0$. In turn, constraint (16) prohibits to start the service at node i after the allowed latest time b_i by setting $\Delta b_i = 0$. In the STW case, time window constraints may be violated at a finite cost (ρ_i) and the vehicle may start the service at node i before time a_i . In such a case, variables Δa_i and Δb_i stand for the size of time windows violations and are determined by expressions (15) and (16), respectively.

Constraint (17) applies in case the maximum allowed working time tv_v^{\max} is regarded as a soft constraint that may be violated at some penalty cost (ρ_v). Otherwise, $\Delta T_v = 0$ and TV_v should not be greater than tv_v^{\max} .

Expression (18) ensures that the current load at customer i never exceeds assigned vehicle's capacity. This constraint only becomes active in pick-up locations, since demands at drop-off nodes are non-positive. Therefore, if a pick-up location with an associated positive load fulfils expression (18), it is also accomplished by consequent delivery destinations.

Constraint (19) allows the current load to be traced at each visited customer. This expression states that it should be greater than, or equal to, the sum of all previous visits and the corresponding demand (w_i). A boolean variable (B_{ij}) is introduced to determine whether a customer is visited previously in the same route or not. Its value is given by the expression

$$B_{ij} \geq B_{ij}^v - S_{ij} \quad \forall i, j \in I, v \in V \quad (20)$$

where B_{ij}^v is determined according to

$$B_{ij}^v \geq Y_{iv} + Y_{jv} - 1 \quad \forall i, j \in I, v \in V \quad (21)$$

3. HYBRID APPROACH

The model presented in the previous section has been programmed using the software ECL¹PS^e. It has been implemented using *eplex* and *ic* libraries, both included in the ECL¹PS^e package. The *eplex* library provides an interface to use an external mathematical programming (LP, MIP or quadratic) solver from within ECL¹PS^e. In this particular case, only default CPLEX solver (ILOG 2001) has been used. The *ic* library contains a general interval propagation solver which can be used to solve problems over both real and integer variables. Therefore, constraint propagation mechanisms are included in this library.

As a previous stage, complete CVRP, VRPTW and PDTW models (i.e. treating allocation and routing subproblems together) have been defined in *eplex* and *ic*, independently. However, separated implementations are not able to solve some problems or have proved to

be quite inefficient. Thus, a hybrid approach has been adopted to tackle the PDTW problem, in order to increase efficiency but loosing optimality.

In the final implementation, different solvers have been used to tackle each subproblem. First, *ic* has been used to solve the allocation subproblem. Second, *eplex* solver is used in the routing subproblem. Both solvers share the allocation variable Y_{iv} and interact according to patterns described on (Apt and Wallace 2007). However, algorithm's structure allows swapping solvers with few changes. *ic* and *eplex* could be easily interchanged, so *ic* would be used to solve the routing subproblem and *eplex* would provide an optimal solution for the allocation step. With this purpose, only expression (2) should be changed to its linear version (7). *ic* functions permit to solve the proposed routing subproblem formulation, but lacking a good efficiency. Moreover, incorporating different algorithms or solvers to the model would also lead to few changes in models' general structure and problem's formulation.

Hybrid algorithm's structure proposed to solve PDTW problems is summarized next:

1. *ic* solves the allocation subproblem
2. *eplex* solves the routing subproblem using Y_{iv} values found by *ic*
3. Check time windows:
 - (a) If they are not fulfilled:
 - (i) Add new constraint to allocation subproblem
 - (ii) Go to step 1
4. Return solution found

It can be observed that, even though the routing subproblem is defined using STW, hybrid algorithm's behaviour corresponds to a HTW model. Time windows constraints are checked after solving the routing subproblem. If they are not fulfilled for a particular route, a new constraint is added to the allocation subproblem, so all customers originally assigned to that route cannot be allocated together in the next iteration:

$$atmost(n_v - 2, Y_{rv}, 1) \quad (22)$$

This constraint states that at least a pick-up/delivery pair has to be removed from vehicle's v route (r_v). This restriction is performed by forcing Y_{iv} variables ($Y_{rv} = \{Y_{iv} | i \in r_v, \subseteq I\}$) to take a maximum of $n_v - 2$ times the value 1, being n_v the current number of customers assigned to route r_v . Expression (22) is a CP global constraint equivalent to the linear expression (23), used in case solvers are swapped.

$$Y_{av} + Y_{bv} + \dots + Y_{lv} \leq \sum_{i \in r_v} Y_{iv} - 2 \quad \forall a, b, \dots, l \in r_v \quad (23)$$

Allocation constraint (22) may be added according to two different criteria. First, it may be added whenever a vehicle has to wait before starting the

corresponding service (arriving to a customer before the lower time window bound a_i) or finishing too late (after the upper bound b_i). On the other hand, constraint (22) may be only added if a vehicle violates any upper bound among customers included in its route. In this case, routes violating only lower bounds, i.e. vehicles are forced to wait but they can perform the service within defined time windows, are supposed to be able to accept new work orders to fill time gaps, so no additional constraints (22) are considered.

Once constraint (22) is added, *ic* solver is called again to rearrange customers. Then, *eplex* optimizes routes using new Y_{iv} values. The process finishes whether a solution fulfilling time windows constraints is found or *ic* solver fails to find a feasible solution for the allocation subproblem. Thus, this algorithm only succeeds if a solution for the HTW case exists. However, the algorithm may be easily modified to track results at each iteration, so it can return a previous solution if it fails.

It should be noticed that, every time a new constraint (22) is added to the allocation subproblem, a pair of pick-up/delivery locations is to be removed from the set of customers assigned to a route. However, it does not specify which particular work order has to be removed from the route. This fact, combined with the simplicity of the objective function (1), may lead to a lack of efficiency due to a random selection of customers to be removed. Some future work could be addressed to define a different objective function or heuristics for the allocation subproblem, in order to reduce this problem's impact.

3.1. Time windows pre-processing

Aiming to increase algorithm's efficiency, time windows defined for each customer are pre-processed to add additional constraints to the allocation subproblem. Depending on the problem instance, some time windows may be overlapped, so a vehicle cannot serve those customers without violating their respective working times. Time windows pre-processing is used to detect this situation and add constraints defining which customers are incompatible to be in the same route.

Two customers are determined to be incompatible if a vehicle cannot perform both services within their respective time windows. Considering a vehicle arriving to customer $i \in I$ and starting the service at a time corresponding to the lower bound of its time window (a_i), it cannot be visited first in the same route than a second customer $j \in I$ if the time cost to reach j from i exceeds former's time window upper bound (b_j), i.e. $a_i + st_i + c_{ij} > b_j$. If this expression also holds swapping indexes i and j , then customers i and j cannot lead to a feasible solution fulfilling time windows when they are included in the same route. Therefore, an additional constraint (24) is defined in the allocation subproblem, so i and j cannot be assigned to the same vehicle v (i.e. $Y_{iv} + Y_{jv} \leq 1 \quad \forall v \in V$):

$$atmost(1, [Y_{iv}, Y_{jv}], 1) \quad \forall v \in V \quad (24)$$

Constraint (24) is defined for each pair of incompatible customers. Thus, a first approach of the number of vehicles and routes to be calculated is obtained by classifying customers into groups where time windows constraints can be fulfilled. Moreover, time windows pre-processing helps to avoid the exploration of some unfeasible solutions, since it provides a more constrained allocation subproblem. Indeed, in the best case it may provide allocation subproblem's solution.

3.2. Routing subproblem decomposition

Routing subproblem is tackled using allocation variable Y_{iv} values obtained from the allocation subproblem. Although customers may be assigned to different vehicles, i.e. different routes, the optimization process used in the presented hybrid model performs the calculation over all routes at the same time. Thus, the number of variables that *eplex* solver should take into account may grow according to problem's instances. This fact may increase dramatically the computation time. Furthermore, time windows violations can only be detected at the end of the process.

In order to reduce the computation time, the proposed routing problem decomposition provides a method to calculate routes separately. With this purpose, an independent routing subproblem is solved for each vehicle, reducing problem's dimensions and improving algorithm's computation time. Moreover, time windows can be checked after each route is calculated, so a partial state that leads to an unfeasible global solution can be detected. Therefore, the routing subproblem decomposition may avoid the exploration of unpromising regions.

A set of customers is assigned to a vehicle v in the allocation process, defining Y_{iv} values. Then, distance and time submatrices can be obtained, including only those customers assigned to a particular vehicle and defining a smaller problem instance. On the other hand, routing subproblem's objective function (8) has been modified to take into account a single vehicle. Thus, the original routing subproblem is decomposed into several smaller instances according to the number of routes to be calculated, reducing the required computation time. Eventually, the total cost associated to the routing process corresponds to the sum of all vehicles' objective function values.

Furthermore, expressions (9)-(18) are simplified by removing all references to Y_{iv} values, since only assigned customers are considered (i.e. $Y_{iv} = 1$ always). Therefore, the number of constraints to be evaluated every time variable S_{ij} is labelled is clearly reduced, increasing algorithm's efficiency. Indeed, constraint (19) can be modified so it only depends on precedence variable S_{ij} values:

$$Q_i \geq w_i + \sum_{j \in I} w_j (1 - S_{ij}) \quad \forall i \in I \quad (25)$$

Time windows can be checked after each route calculation or after all routes are determined. The former may save computation time, since all routes may not have to be calculated at each iteration. The latter may not provide such a computation time reduction, but a complete solution may be obtained at each iteration. In both cases, an additional constraint (22) is added to the allocation subproblem for every route whose assigned customers' time windows are violated.

In the proposed model, time windows are checked after all routes have been calculated. Performance is improved due to smaller instances are tackled. At the same time, a complete solution is calculated at each iteration, even though time windows are not fulfilled, so a response may be provided if the allocation process fails after adding new constraints (22). Considering time windows checking after each route calculation would make this process more complex, since some routes would have to be recalculated after the allocation process fails.

4. APPLICATION AND RESULTS

The model presented in previous sections has been tested on small PDTW problem instances. These test cases have been defined aiming to detect errors and ensure all implementations provide expected results. Finally, a real-case based problem (Busquets et al. 2005) with a reasonable number of customers and work orders has been defined. Unfortunately, the model could not be tested on PDTW benchmark problems, since there is not a set of well accepted benchmark instances, although some efforts have been addressed in this direction (Lau and Liang 2001).

PDTW instances defined to test algorithm's validity have the following characteristics:

- 6 customers grouped in 3 work orders, 1 depot and 2 vehicles with different capacities. Optimal solution's cost is 483 and 2 vehicles are needed.
- 7 customers grouped in 5 work orders, where 4 of them share the same pick-up location, 1 depot and 2 vehicles with different capacities. Optimal solution's cost is 266 and 1 vehicle is needed.
- 33 customers grouped in 16 work orders, where pick-up locations are shared, 1 depot and 3 vehicles with the same capacity. Optimal solution's cost is 2825.9 and 3 vehicles are needed.

Table 1 show results obtained for different implementations and solvers combinations. In all cases, obtained solutions provide the exact number of vehicles required in the optimal case. Deviation from optimal solution is only indicated if the algorithm has not reached the optimum, otherwise it is 0%. Blanks correspond to those cases where the algorithm could not find a solution in a reasonable amount of time.

Table 1: Results Obtained for PDTW Instances

Solver	6 cust.	7 cust. same pp	33 cust.	
	Sol.	Sol.	Sol.	Dev.
eplex	483 (0.17s)	266 (1.50s)	-	-
ic	-	266 (3.02s)	-	-
eplex-ic	483 (0.23s)	266 (0.31s)	-	-
eplex-ic + TWP	483 (0.13s)	266 (0.34s)	-	-
eplex-ic + routes	483 (0.19s)	266 (0.47s)	-	-
eplex-ic + TWP + routes	483 (0.14s)	266 (0.44s)	-	-
ic-eplex + TWP	483 (0.16s)	266 (0.25s)	3134.1 (9.23s)	11%
ic-eplex + TWP + routes	483 (0.08s)	266 (0.33s)	3134.1 (8.59s)	11%

First two rows present results for single-solver initial models. It can be observed that *eplex* is more efficient than *ic*'s implementation, due to the model is more suitable for a MILP solver and does not correspond to a CP specific formulation. Next rows show results obtained for the proposed hybrid model combined with presented performance improvements, i.e. time windows pre-processing and routing subproblem decomposition. Several tests have been done interchanging solvers' tasks. First four results correspond to using *eplex* as allocation subproblem's solver and *ic* to tackle the routing task. Reverse assignation achieve results presented in last rows. It can be noticed that all models reach the optimum for small PDTW instances. On the other hand, 33-customers problem is only solved by *ic-eplex* hybrid algorithms, although the optimal solution is not reached. Computation times are also shown, with lowest values highlighted for each problem. In all cases, best calculation times are obtained for the *ic-eplex* model described in the present paper. It should be remarked that, in the 7-customers problem with shared pick-up locations, neither time windows pre-processing nor routing subproblem decomposition provide a significant computation time decrease. In this particular case, a single vehicle is needed to serve all customers, so only one route has to be configured. Thus, time may be spent checking corresponding time windows without adding additional constraints. Moreover, despite all models have to calculate a single route including same customers, routing subproblem decomposition forces the algorithm to dynamically define the *eplex* instance after solving allocation subproblem, i.e. additional time is to be spent on checking variables' values.

5. CONCLUSIONS AND FUTURE RESEARCH

The present paper introduces the developed study about the PDTW, a well known NP-Hard problem. It presents a complete model, decomposed into two subproblems based on CP and MILP paradigms, to make it suitable to adopt a hybrid approach.

Model's implementation has been described, as well as general interaction patterns between solvers

used to tackle the problem: *ic* and *eplex*. Moreover, performance improving techniques have been introduced, such as time windows pre-processing and routing subproblem decomposition. This hybrid approach has proved to be more efficient than those based on a single solver. In addition, algorithm's structure and problem's definition permit to easily interchange or substitute these solvers. Thus, some future efforts could be focused on developing specific methods, such as Lagrangian Relaxation, to tackle each subproblem more efficiently.

This approach has demonstrated to be suitable to solve different problems in a reasonable computation time. However, the algorithm may not guarantee finding the optimal solution. Reasons might be found in subproblems decomposition, as well as constraint addition mechanisms. Further research is to be addressed in these directions to improve solutions' quality while keeping a low calculation time.

REFERENCES

- Apt, K. and Wallace, M., 2007. *Constraint Logic Programming using ECLiPS^e*. Cambridge University Press.
- Bräysy, O. and Gendreau, M., 2005. Vehicle routing problem with time windows, part I: route construction and local search algorithms. *Transportation Science*, 39: 104-118.
- Bräysy, O. and Gendreau, M., 2005. Vehicle routing problem with time windows, part II: metaheuristics. *Transportation Science*, 39: 119-139.
- Busquets, S., Vilalta, L., Piera, M.A., Guasch, T. and Narciso, M., 2005. Specification of metaheuristics in colored petri nets models to tackle the Vehicle routing problem. *Proceedings of the International Mediterranean Modelling Multiconference (IM'05)*. October 20-22, Marseille (France).
- Dondo, R. and 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.
- ILOG, 2001. *ILOG CPLEX 7.1 User's Manual*. ILOG France.
- Lau, H. and Liang, Z., 2001. Pickup and delivery with time windows: algorithms and test case generation. *Proceedings of IEEE International Conference on Tools with Artificial Intelligence (ICTAI'01)*, pp.333-340. November 7-9, Dallas (Texas, USA).
- Savelsbergh, M., 1985. Local search in routing problems with time windows. *Annals of Operations Research*, 4: 285-305.
- Savelsbergh, M. and Sol, M., 1995. The general pickup and delivery problem. *Transportation Science*, 29: 17-29.

AUTHORS BIOGRAPHY

Daniel Guimarans is a PhD Student and Assistant Teacher at the Telecommunication and System Engineering Department at the Universitat Autònoma de Barcelona. He is an active member of the Logisim Research Group. His research interests are constraint programming, simulation and hybridisation of different techniques to solve industrial combinatorial problems.

Juan José Ramos is an Associated Professor at the Telecommunication and System Engineering Department at the Universitat Autònoma de Barcelona. He is coordinator and an active member of the Logisim Research Group. His research interests are modelling, simulation and optimisation of logistics problems.

Mark Wallace holds a chair in the Faculty of Information Technology at Monash University. He is director of the CTI-Monash Centre for Research in Optimisation and belongs to the Centre for Research in Intelligent Systems and the Optimisation and Constraint Solving Research Group. He is also involved in the NICTA Constraint Programming Platform project ("G12"). His research interests are focused on constraint programming, hybridisation of different techniques and its application to industrial combinatorial optimisation problems.

Daniel Riera received in 2006 his Computer Science PhD in the Universitat Autònoma de Barcelona. He is director of the Computer Science department in the Universitat Oberta de Catalunya and researcher of the GRES-UOC. Currently, he is co-director of the HAROSA (Hybrid Algorithms for solving Realistic rOuting, Scheduling and Availability problems) knowledge community. His research interests are constraint programming, simulation and techniques hybridisation to solve industrial combinatorial problems.

A HYPER-HEURISTIC APPROACH FOR MODELING ASSIGNATION PRIORITIZATION RULES ON VRPTW CONSTRUCTIVE HEURISTIC BY NEURAL NETWORKS.

Diego Crespo Pereira^(a), David del Río Vilas^(b), Alejandro García del Valle^(c), Adolfo Lamas Rodríguez^(d)

^(a) ^(b) ^(c) ^(d) Grupo Integrado de Ingeniería, University of A Coruña (Spain)

^(a) dcrespo@udc.es, ^(b) daviddelrio@udc.es, ^(c) agvalle@udc.es, ^(d) alamas@udc.es

ABSTRACT

Heuristic constructive algorithms have been widely and successfully applied to the solution of routing problems. These algorithms generally consist of a sequential or parallel assignment of nodes to constructed routes. Prioritization rules for assignments are critical for a good performance of the algorithm. This paper presents a hyper-heuristic technique based on a neural network model for prioritization rules which is evolved using an evolutionary search. The technique is applicable to general routing problems, although this paper focuses on the VRPTW. A set of factors specific about the VRPTW for assignment evaluation is introduced. The algorithm is tested with the Solomon instances benchmark with tests oriented to the performance evaluation and the generalization capability of the evolved neural network when solving problems for which has not been evolved.

Keywords: Hyper-heuristic, prioritization rules, VRPTW, neural network.

1. INTRODUCTION

Routing problems constitute one of the most well known and characteristic problems in operations research. They play a central role in logistics due to the relatively high importance of distribution costs in supplying chain (Figliozzi 2007).

In general terms, routing problems deal with finding an optimal or quasi-optimal set of routes for rendering the corresponding services - freight transportation, repair services, passengers transportation, etc.- to a given set of customers or suppliers geographically distributed and connected by a transportation network. But most of routing problems are NP-complete, which causes high computational costs when aiming to find optimal solutions. This has led to a widespread development of heuristic and metaheuristic approaches, more suitable for real applications.

Places where to render services and pick up or deliver freights are modeled as vertices in a graph where edges are the paths in the underlying transportation network. Different sets of constraints and types of graphs lead to the numerous specific problems in routes optimization. Some of the most classical and studied ones include the TSP, the CVRP or the one in

which this work has focused, the VRPTW (Solomon and Desrosiers 1988). An important division between types of routing problems is set up by the dynamic or static nature of the problem (Larsen 2000). A problem is said dynamic if information is not completely known a priori, when the routes construction is initiated, and this information is completed as vehicles cover the routes.

Common approaches to solve routing problems include optimal algorithms, heuristic algorithms and metaheuristics. Optimal algorithms are non-polynomial time solving –and thus not so appropriate for practical applications. Heuristic algorithms exploit appropriate knowledge about the problem features and are able to find almost optimal solutions with very competitive time consumption. However, they have the drawback of depending on problem features and thus it is possible to find instances for which performance is poor. Finally, metaheuristics are usually not as competitive as the best heuristics, but more general in their definition and thus more easily adaptable to problems with different nature.

Currently, main trends in routes optimization research include the definition of new problems closer to applications in real environments (Van Woensel 2008), research of new hybrid algorithms with higher performance and generalization capabilities (Chen 2005) and the improvement of existing solution techniques.

In this paper, a hyper-heuristic technique (Bai 2007) is introduced in order to develop a suitable algorithm for general routing problems in real environments. This technique is applied to the solution of the VRPTW as an initial study to evaluate its feasibility and to identify the main aspects for its future improvement.

2. HEURISTICS CONSTRUCTIVE ALGORITHMS FOR ROUTING PROBLEMS

Construction algorithms for routing problems consist of a sequential construction of the solution routes so that in every step one or a certain number of nodes are inserted into partially obtained routes based on a given criteria. Common criteria might be regarded as the selection of a local optimal insertion among a set of feasible insertions. Every insertion performed induces a change on the feasible insertions on the following steps, so that local optimal insertion decisions do not necessarily lead to global optimal solutions. An insertion algorithm that

only attempts to select the insertion with lowest increase on total distance of routes is a greedy algorithm. The well known problem caused by this approach is that on the first steps of the algorithm low distance insertions are selected with the eventual cost of leaving many nodes for insertion in inappropriate routes during the last steps of the algorithm.

To face this problem, most of insertion algorithms include criteria to select insertions based on more complex rules. Designed rules should provide an estimate not only for local making decision cost, but an estimation of the impact in global solution. For instance, in VRPTW, waiting time of inserted node is often considered due to the opportunity cost of being waiting to serve a node instead of serving others.

Solomon (1987) developed a heuristic for the VRPTW in which three criteria are used for the insertion selection. These include 1) Latest time for returning to the depot, 2) Lowest gap between service beginning time for a customer and customer closing time and 3) a pondered insertion gain that is based on many factors about the vehicle, node and restrictions.

Faulín and García del Valle (2007) developed an heuristic algorithm for the CVRP in which the criteria followed to sort the assignments to the routes is based on an analogy to the electrostatic charges attraction. Thus, attraction force between nodes is calculated and used to prioritize those with highest attraction.

Generally, a construction algorithm starts with a set of nodes unassigned to routes (or the equivalent situation: assigned to a route in which only that node is serviced). Then, iteratively, nodes are added to construction routes according to the prioritization rules. The order in which assignments are performed is critical since selection implies that other assignments become infeasible in the following steps. For instance, if a node with a high demand is assigned to an almost full route, no other node will be allowed in the same route due to capacity restrictions. Another example is the situation of two nodes A and B for which node C is the closest. If node A is assigned to a construction route that ends in C, then B is not allowed to be assigned to its closest in following iterations.

These prioritization rules are determinant for heuristic algorithms. In this paper we propose the substitution of predefined rules for a parameterized set of rules based on a wide set of factors specific about the routing problem.

3. ALGORITHM DESCRIPTION

In order to describe our algorithm, we first introduce the following notation. This notation is used for the VRPTW

Let $G = \{V, E\}$ be the graph of the transportation network where V denotes the set of nodes. v_0 is the depot. A route is a cycle in V that contains the depot:

$$R = \{v_0, a_{0,1}, v_1, a_{1,2}, \dots, v_{|R|-1}, a_{|R|-1,|R|}, v_{|R|}, a_{|R|,0}; v_i \in V; a_{i,i+1} = (v_i, v_{i+1})\}. \quad (1)$$

The distance of an edge will be a function $d: E \rightarrow \mathbb{R}$ such that $d_{ij} = d(v_i, v_j)$. No further assumption is made on distances function, thus this algorithm might be either applied to cases with Euclidean, symmetric or asymmetric distances. The distance of a route is the sum of its edges distances:

$$d(R_k) = \sum_{i=0}^{|R|-1} d(v_i, v_{i+1}) + d(v_{|R|}, v_0); v_i \in R \quad (2)$$

The travelling time from one node to another is also a function $t: E \rightarrow \mathbb{R}$ such that $t_{ij} = t(v_i, v_j)$. This function has to be positive for every edge, but we do not assume any further restriction or relation with the distance. This let us model routing problems for which the average traveling speed is different for every arc.

Given a route, the sequence of nodes visited from the depot defines an arrival (at) and a departing (dt) time for each node depending on its opening time (ot) and service time (st) as follows:

$$at_i = dt_{i-1} + t_{i-1,i} \quad (3)$$

$$dt_i = \max\{at_i + st_i, ot_i\} \quad (4)$$

A solution to the routing problem is a set of routes S that verifies the following constraints:

1. $\forall v \in V \rightarrow \exists R \in S$ such that $v \in R$.
2. $R_i \cap R_j = \{v_0\} \forall i \neq j$.
3. The load of a vehicle cannot exceed its capacity C . In this paper, we assume all the vehicles to have the same capacity. Thus, $\forall R_k \in S \rightarrow \sum_{i=1}^{|R|} q_i \leq C; q_i \in R_k$.
4. The arrival time to a node cannot exceed its closing time (ct). $at_i \leq ct_i \forall i \in S$.

Then, the VRPTW aims at finding the solution S^* that minimizes the sum of distances of its routes. Let Ψ be the set of all the solutions to the problem. Thus:

$$\sum_{R \in S^*} d(R) \leq \sum_{R \in S} d(R) \quad \forall S \in \Psi \quad (5)$$

An assignment is defined as a pair node-route. $A_{ki} = A(R_k, v_i) = \{R_k, v_i | v_i \notin R_k\}$. Every assignment defines an extended route of R_α as the route formed by adding the node at the end of the route. $R'_\alpha = \{v_0, v_{k1}, \dots, v_{kn}, v_i; v_{kj} \in R_k\}$. The cost of an assignment is a function that represents the difference between the distance of the initial and extended routes. $CF(A_{\alpha i}) = d(R'_\alpha) - d(R_\alpha)$. Every route can thus be obtained as a sequence of assignments starting from a route that only contains the depot.

$$R_\alpha = \{v_0, v_{\alpha 1}, \dots, v_{\alpha m}\} = \{A(\emptyset, v_{\alpha 1}), \dots, A(\{v_{\alpha 1}, \dots, v_{\alpha m-1}\}, v_{\alpha m})\} \quad (6)$$

An assignment is said feasible if the extended route verifies the problem constraints. It can be easily proved

that a route is feasible if and only if all the assignments necessary to obtain it are so.

Since every route can be decomposed into a sequence of assignments, a solution can be also regarded as a set of feasible assignments. A partial solution is a set of assignments such that at least one solution exists containing all of them. The feasibility of an assignment given a partial solution depends on the latter one.

The constructive algorithm that we propose consists of an iterative performance of assignments such that the obtained solution is intended to have a distance close to the optimal solution S^* . On the n -th step, there is a set of feasible assignments (FA_n) that can be added to the temporary partial solution constructed and one of them has to be performed. Thus, the key of the algorithm are the rules followed to determine which one of the feasible assignments is chosen. These prioritization rules can be modeled as a function of a set of factors that characterize the assignment: $p_{ki} = PR(f_1(A_{ki}), \dots, f_{NF}(A_{ki}))$, where $f_j(A_{ki})$ is each one of the assignment factors and NF the total number. Ties in priority might be broken randomly or by a proper selection of the model.

In order to increase algorithm's speed, the priority is not calculated for all the feasible assignments. A first selection of studied assignments (SA_n) is made according to one of the factors, the selecting factor (SF). Only the NK assignments with highest SF to each route in the partial solution are selected for prioritization.

The procedure is the following, where:

*Let L_0 be the set of not yet assigned vertices.
PS is the partial solution.*

1. $PS = \{v_0\}$.
2. $L_0 = V - \{v_0\}$.
3. Obtain $SA_{n,1}$.
4. $j = 1$.
5. Calculate priorities for routes in $SA_{n,j}$ by the prioritization model.
6. Search the assignment $A_{i^*} \in SA_{n,j}$ such that $p_{i^*} \geq p_i; A_i \in SA_{n,j}$.
7. If the route $R_{i^*} \in A_{i^*}$ only contains the depot $R_{i^*} = \{v_0\}$ then create a new route containing only the depot.
8. Add vertex $v_{i^*} \in A_{i^*}$ to route R_{i^*} .
9. Remove v_{i^*} from L_0 .
10. Remove from $SA_{n,j}$ all the assignments that contain either the node or route of A_{i^*} : $SA_{n,j+1} = SA_{n,j} - \{A_{k,i} | (k = k^*) \cup (i = i^*)\}$.
11. If $j \leq ND$, then go to step 4 and make $j = j + 1$. Otherwise continue.
12. If $L_0 \neq \emptyset$ return to 3.

A group of assignment factors has been defined based on VRPTW properties judged relevant. In order

to introduce these parameters, we first introduce the following notation:

Given a sorted list L and its element e_i , then the order of the element is its position in the list sorted from minimum to maximum and it is noted as $O(e_i, L)$.

The assignment with subscript a for which the parameters are defined is the pair route-node (R_k, v_j) .

The route head node (RH) is the last node of a given route.

The direct gain of an assignment is the gain obtained by performing an assignment compared to the second best possible assignment to the route. Thus:

$$dg_{k,i} = \min_j \{d_{RH_k,j}; d_{RH_k,j} \neq \min_l \{d_{RH_k,l}\}\} - d_{RH_k,i} \quad (7)$$

The total gain of an assignment is the direct gain obtained by performing an assignment compared to the second best direct gain to assign the node.

$$g_{k,i} = dg_{k,i} - \max_j \{dg_{RH_j,i}; dg_{RH_j,i} \neq \max_l \{dg_{RH_l,i}\}\} \quad (8)$$

The normalization waiting factor (WNF) is used to normalize waiting times and can be set up according to characteristic time units in the problem.

Direct degree of a node $\delta(v_i)$ is the number of feasible assignments to a route ended by it.

Inverse degree of a node $\delta'(v_i)$ is the number of feasible assignments to a route ended by it.

The selection factor used in our algorithm is the total gain. Thus, in the SA_n calculation step, the total gain is evaluated for each assignment in (FA_n). For each route in the partial solution, the first NK assignments with highest total gain are added to SA_n . The nodes that can only be accessed from the depot are also added to SA_n .

The rest of the factors for the prioritization rules model have been chosen with the aim of providing enough information about the assignment to make possible good prioritization rules. The parameters are transformed into normalized values between 0 and 1. Thus they do not depend on problem size or units.

For the VRPTW we employ the following sixteen factors:

1. Relative total gain between the maximum and the minimum gain for all the selected assignments. If no more than one assignment is feasible for a node or a route, its value is assumed to be 1. Expected behavior for evolved rules is to prioritize routes with high gains. In case of no difference between top and lowest gain, this factor equals 1.

$$RTG_{k,i} = \frac{g_{k,i}}{\max_{A \in LA} g(A)} \quad (9)$$

2. Relative total gain to route. It is the relative total gain between the maximum and the minimum gain

for all assignments which correspond to the same route as evaluated assignment. If no more than one assignment is feasible for the route, its value is assumed to be 1. Expected behavior for evolved rules is to prioritize routes with high gains. In case of no difference between top and lowest gain, this factor equals 1.

$$RGR_{k,i} = \frac{g_{k,i}}{\max_j g_{k,j}} \quad (10)$$

3. Waiting factor at the visited node. It is defined to prioritize assignments that do not cause high waiting times in the route.

$$wt_{k,i} = \begin{cases} \frac{1}{1 + \frac{t_{o_i} - t_{a_i}}{WNF}} \\ 1 \text{ otherwise} \end{cases} \quad (11)$$

4. Arrival time factor. The aim of this factor is to provide different prioritization rules based on whether is a node early assigned to route or lately.

$$TAF_{k,i} = \frac{t_{a_i}}{\max_j t_{c_j}} \quad (12)$$

5. Centrality factor. It is the relative distance from node to depot related to the highest distance to depot.

$$CNF_i = \frac{d_{0,i}}{\max_j d_{0,j}} \quad (13)$$

6. Centrality order. It is the position in the ordered list of distances to depot. Together with centrality factor is expected to provide combined prioritization rules as a function of the proximity to depot compared to the rest of nodes in the problem.

$$CNO_i = O(d_{0,i}, \{d_{0,j}; j \in 1, \dots, N\}) \quad (14)$$

7. Closing time factor. It is the relative closing time compared to the latest closing time in the problem.

$$CTF_i = \frac{t_{c_i}}{\max_j t_{c_j}} \quad (15)$$

8. Closing time order. It is the position in the ordered list of closing times to depot. Together with closing time factor is expected to provide combined prioritization rules as a function of the closing time compared to the rest of nodes in the problem.

$$CTO_i = O(ct_i, \{ct_j; j \in 1, \dots, N\}) \quad (16)$$

9. Inverse degree factor. It is defined as the fraction of non assigned nodes from which assignment node can be accessed.

$$IDF_i = \frac{\delta'(i)}{N} \quad (17)$$

10. Direct degree factor. It is defined as the fraction of non assigned nodes that can be accessed from assignment node.

$$DDF_i = \frac{\delta(i)}{N} \quad (18)$$

11. Assignment demand factor. It is defined as the degree of completeness of the vehicle capacity if the node was assigned to the route.

$$ADF_{k,i} = \frac{\sum_{j \in R_k} q_j + q_i}{C} \quad (19)$$

12. Assigned rate. It is the fraction of nodes that have already been assigned at the considered step. It is expected to provide different rules depending on the algorithm completion rate.

$$AR = \frac{|L_0|}{N} \quad (20)$$

13. Centrality factor of route. It is the relative distance from route head to depot related to the highest distance to depot.

$$CNFR_{CR_k} = \frac{d_{0,CR_k}}{\max_j d_{0,CR_k}} \quad (21)$$

14. Centrality order of route. It is route head position in the ordered list of distances to depot.

$$CNOR_{CR_k} = O(d_{0,CR_k}, \{d_{0,CR_k}; j \in 1, \dots, N\}) \quad (22)$$

15. Route load factor. It is the fraction of vehicle capacity already served by the route.

$$RLF_k = \frac{\sum_{j \in CR_k} q_j}{C} \quad (23)$$

16. Direct degree fraction of the route. It is the fraction of non assigned nodes that can be accessed from the route.

$$DDF_k = \frac{\delta(CR_k)}{N} \quad (24)$$

Our approach consists of defining a parameterized prioritization function model which to optimize. Thus the routing problem is turned into a parameters (γ) optimization problem which is solved by means of a search algorithm. The objective function for the search algorithm is the total distance of the solution obtained for a given set of parameters.

$$p_{ki} = PR(f_1(A_{ki}), \dots, f_{16}(A_{ki}), \gamma_1, \dots, \gamma_{NP}) \quad (25)$$

NP is the number of parameters in the model.

It can be defined a function of the parameters on the solutions set such that:

$$d(\gamma_1, \dots, \gamma_{NP}) = d(S(\gamma_1, \dots, \gamma_{NP})) \quad (26)$$

This is the function we attempt to minimize by finding the parameters vector.

4. PRIORITIZATION RULES MODEL

The model considered for the prioritization rules is a neural network. Specifically, the neural network used is a multilayer perceptron with sigmoid activation function due to its capability to approximate every real function, including non continuous functions if it has more than one hidden layer (Hornik 1989). The multilayer perceptron used has one hidden layer with 4 neurons. This means that a set of 78 parameters are needed to define the net weights, bias and slopes. Maximum and minimum weight values and bias are limited to 8 and -8 respectively. The slopes are in the range from 0.2 to 1.8.

In order to optimize neural network parameters an evolutionary strategy is used. Genes in the chromosome are the net weights, bias and slopes. Also the number of assignments selected for each route (NK) and the number of performed assignments (ND) are included in the chromosome. The objective function is the total distance calculated solving a problem or a set of problems. Notice that once a priority model is evolved, it can be used to solve problems different to the evolving set. The capability to obtain good solutions when solving problems different to the evolving set is the most desired feature of this technique.

Two search algorithms have been used in this work. One is the Differential Evolution strategy as defined in (Rainer Dorn and others, 1997). The parameters used for the evolution are $F = 0.7$ and $CR = 0.3$. Since it is an algorithm to evolve genes being real numbers, ND and NK are coded as real numbers between -1 and 1. Then they are transformed into integer values by setting top and lowest values.

The other search algorithm employed is the Macroevolutionary Algorithm (MA) developed by J. Marín and R. V. Solé (Marín 1999). Both evolutionary strategies are compared and the MA is selected due to its highest performance.

Each objective function for the evolutionary algorithm is the distance of the solution to a VRPTW instance by using the algorithm described in the previous section. In each evaluation, the neural network employed is the given by the individuals chromosome coding.

5. EXPERIMENTATION

The algorithm has been coded in JAVA and makes use of the Evolutionary Algorithm Framework Library and the Artificial Neural Network Library developed by the Grupo Integrado de Ingeniería of the University of A Coruña.

As benchmark of problems to evaluate algorithm's performance, we employ the Solomon's Instances. In order to test the algorithm's capability to find rules suitable for many problems, a subset of these instances was used for evolving the neural networks. It is the set of problems derived from the Solomon's benchmark by selecting only the first 26 nodes in each one. As

benchmark for testing the neural networks evolved, we employ the 26, 51 and 101 nodes instances. The best solutions to this benchmark can be found in Solomon's website. The performance measure employed is the relative increase in distance to the optimal solution.

Four types of tests have being conducted in order to test algorithm's performance. The first is to compare both evolutionary strategies by evolving a neural network for each problem in the Solomon's 26 nodes instances. Results presented in table 1 show that the MA's performance is higher. In both cases the stop criteria for the algorithm is the same: 10,000 objective function evaluations as a top. The difference is maximum on random cluster type problems and minimum on cluster.

Table 1: Comparison between MA and DE.

26 Nodes Solomon's Instances				
Problem	MA	DE	Rel. Diference	Max. Rel Diference
R1	5,995	6,247	4.20%	8.53%
R2	4,615	4,938	7.00%	11.38%
C1	1,759	1,768	0.51%	4.31%
C2	1,733	1,743	0.58%	3.70%
RC1	2,989	3,106	3.90%	12.74%
RC2	2,846	3,191	12.13%	27.38%
Total	19,937	20,992	5.29%	27.38%

The second test attempts to find the best solution that can be obtained by evolving a neural network for each problem. Thus, a neural network is evolved for every Solomon's instance and compared to the optimal solution. This is accomplished for 26 nodes instances by means of the MA and setting the maximum number of objective function evaluations to 60,000. The population size is 1,500 and the number of generations 40. Results obtained are good since *quasi*-optimal solutions are found (table 2). Algorithm's performance is higher for cluster type problems than for random ones.

Table 2: Best solutions found with the MA.

26 Nodes Solomon's Instances				
Problem	Optimal Distance	Distance	Rel. Diference	Max. Rel Diference
R1	5,560	5,788	4.09%	8.75%
R2	4,204	4,439	5.60%	11.52%
C1	1,716	1,733	0.99%	4.84%
C2	1,716	1,733	1.00%	4.30%
RC1	2,802	2,872	2.51%	3.86%
RC2	2,554	2,690	5.33%	10.54%
Total	18,552	19,256	3.79%	11.52%

The third test consists of using the best neural networks obtained for the 26 nodes instances and apply them to solve the 51 and 101 cases. Thus the generalization capability of the evolved neural networks for solving bigger problems is checked. The results are shown in tables 3 and 4 and show as the average relative distance to the optimum increases with the problem size. The obtained solutions are bad compared

to the solutions achieved for the problem size that they have been evolved for. However, these problems are rather different and hence this result might be expected.

Table 3: Best neural networks for the 26 nodes case applied to 51 nodes instances.

51 Nodes Solomon's Instances				
Problem	Optimal Distance	Distance	Rel. Difference	Max. Rel Difference
R1	9,194	11,768	28.00%	53.83%
R2	6,967	9,219	32.33%	51.39%
C1	3,255	4,853	49.10%	147.62%
C2	2,860	4,063	42.07%	71.48%
RC1	5,852	8,464	44.65%	71.18%
RC2	4,671	7,350	57.36%	78.97%
Total	32,798	45,717	39.39%	147.62%

Table 4: Best neural networks for the 26 nodes case applied to 101 nodes instances.

101 Nodes Solomon's Instances				
Problem	Optimal Distance	Distance	Rel. Difference	Max. Rel Difference
R1	14,146	19,851	40.34%	64.80%
R2	10,362	13,649	31.72%	45.70%
C1	7,440	11,621	56.18%	121.43%
C2	4,699	7,225	53.75%	116.35%
RC1	10,731	16,857	57.08%	110.81%
RC2	8,392	12,325	46.87%	81.10%
Total	55,770	81,527	46.19%	121.43%

The fourth test performed consists of evolving the neural network for each instance with a randomly modified version of it. All the node coordinates, demand values, opening times, service times and closing times are randomly modified with a uniform distribution between its value minus and plus a specified factor. Then the evolved neural network is applied to solve the original problem and thus its capability to solve similar problems can be studied. The factor levels for the modification are set between -5% to 5% and -10% to 10%. The results are shown in tables 5 and 6. The increase in relative gap to optimal solution increases with the modifying level. The performance becomes worse, but it should also be noted that the modified problems are very different to the original ones, since the modifications in coordinates, demands and time windows change the feasible edges between nodes.

Table 5: Neural networks evolved for Solomon's instances randomly modified by a 5%.

26 Nodes Solomon's Instances				
Problem	Optimal Distance	Distance	Rel. Difference	Max. Rel Difference
R1	5,560	7,169	28.93%	46.76%
R2	4,204	5,736	36.47%	60.93%
C1	1,716	2,241	30.58%	73.96%
C2	1,716	2,219	29.36%	69.26%
RC1	2,802	3,925	40.09%	95.04%
RC2	2,554	3,622	41.82%	122.80%
Total	18,552	24,914	34.29%	122.80%

The neural networks for the modified problem are evolved with the MA, a population size of 1,500 individuals and 40 generations.

Table 6: Neural networks evolved for Solomon's instances randomly modified by a 10%.

26 Nodes Solomon's Instances				
Problem	Optimal Distance	Distance	Rel. Difference	Max. Rel Difference
R1	5,560	7,431	33.63%	55.93%
R2	4,204	5,783	37.58%	52.12%
C1	1,716	2,647	54.25%	140.92%
C2	1,716	2,533	47.66%	138.75%
RC1	2,802	4,264	52.15%	113.41%
RC2	2,554	3,847	50.61%	84.50%
Total	18,552	26,505	42.87%	140.92%

6. CONCLUSIONS

A hyper-heuristic algorithm with complex rules for the prioritization of assignments has been introduced for the VRPTW and tested with the standard benchmark Solomon's instances. This algorithm might be either applied to cases with Euclidean, symmetric or asymmetric distances. It also allows model routing problems for which the average traveling speed is different for every arc. The Macroevolutionary Algorithm (MA) has shown a good performance evolving the rules modeled by a multilayer perceptron.

The results are good when solving problems for which the rules have been evolved with the MA. It is able to reach *quasi*-optimal solutions with the advantage that the found solution is not just a set of routes that minimize the distance, but the neural network which contains the constructive routes that lead to them.

Another advantage of the introduced algorithm is that, since few assumptions have been made on problem constraints, it might be applied to VRPTW variations without big effort on adaptation expected.

However, the generalization capabilities of the evolved rules have shown to be poor either at solving problems with bigger size or randomly modified from the original instance. Input factors for the neural network are critic on this matter. If the factors values present different ranges on different instances, the rules evolved for one will perform badly when applied to another. Hence, the proposed group should be carefully studied and improved in order to enhance these results.

ACKNOWLEDGMENT

We wish to express our gratitude to the Xunta de Galicia, which has funded this work through the research project "Análisis y desarrollo de una aplicación para la gestión, planificación y optimización de rutas de transporte en entornos dinámicos" (PGIDIT06DPI000PR10).

REFERENCES

Bai, R., Blazewicz, J., Burke, E.K., Kendall G., McCollum, B. 2007. A simulated annealing hyper-heuristic methodology for flexible decision

support. Technical report, School of CSiT, University of Nottingham, UK.

- Chen, A., Yang, G., Wu, Z. 2005. Hybrid discrete particle swarm optimization algorithm for capacitated vehicle routing problem. *Journal of Zhejiang University - Science A*, 7 (4), 607-614.
- Faulin, J., García del Valle, A. 2007. Solving the capacitated vehicle routing problem using the ALGELECT electrostatic algorithm. *Journal of the Operational Research Society*, 1-11.
- Figliozzi, M.A., Mahmassani, H.,S., Jaillet, P. 2007. Pricing in Dynamic Vehicle Routing Problems. *Transportation Science*, 41 (3), 302-318.
- Hornik, K., Stinchcombe, M., White, H. 1989. Multilayer feedforward networks are universal approximators, *Neural Networks*, 2, 359-366.
- Larsen, A., 2000. The Dynamic Vehicle Routing Problem. Ph.D.-thesis. IMM-PHD-73, IMM.
- Marín, J., Solé, R.V. 1999. Macroevolutionary Algorithms: A New Optimization Method on Fitness Landscapes. *IEEE transactions on evolutionary computation*, 3 (4).
- Rainer, D., Kenneth, P. 1997. Differential Evolution. A Simple and Efficient Heuristic for Global Optimization over Continuous Spaces. *Journal of Global Optimization*, 11, 341-359.
- Solomon, M.M. 1987. Algorithms for the vehicle routing and scheduling problems with time window constraints. *Operations Research*, 35 (2), 254-265.
- Solomon, M.M., Desrosiers, J. 1988. Time window constrained routing and scheduling problems. *Transportation Science*, 22(1), 1- 13.
- Van Woensel, T., L. Kerbache, L., Peremans, H., Vandaele, N. 2008. Vehicle routing with dynamic travel times: A queueing approach. *European Journal of Operational Research*, 186 (2008), 990-1007.

AUTHORS BIOGRAPHY

Diego Crespo Pereira holds an MSc in Industrial Engineering (Ingeniero Industrial) and he is currently studying for a PhD. He works full-time in the GII of the University of A Coruna as a research engineer since 2008. He is mainly involved in the development of R&D projects related to industrial and logistical processes optimization. He is especially interested in the field of dynamic vehicle routing problems.

David del Rio Vilas holds an MSc in Industrial Engineering (Ingeniero Industrial) and has been studying for a PhD since 2007. He works full-time in the research group GII of the University of A Coruna as a research engineer since 2007. He is mainly involved in R&D projects development related to industrial and logistical processes optimization. His area of major interest is the use of simulation in industrial and logistical environments, especially in maritime container terminals.

Alejandro Garcia del Valle is an Industrial Engineer by the Polytechnic University of Madrid and holds a PhD in Industrial Engineering. He is Professor and one of the founders of the GII at the University of A Coruna. He has led many research projects about simulation and optimization of industrial, logistical and service systems.

Adolfo Lamas Rodriguez graduated from the University of Vigo in 1998. He holds an MSc and a PhD in Industrial Engineering. He combines his research activities in the GII and his position as a senior engineer in the Spanish leading shipbuilding company *Navantia*. He is also Associate Professor in the University of A Coruna.

SOLVING THE TRAVELLING SALESMAN PROBLEM WITH TIMEWINDOWS BY LAGRANGEAN RELAXATION

R. Herrero^(a), J. J. Ramos^(b) and D. Guimarans^(c)

Department of Telecommunication and System Engineering
Universitat Autònoma de Barcelona
08193, Bellaterra, Spain

^(a) Rosa.Herrero@uab.cat, ^(b) JuanJose.Ramos@uab.cat, ^(c) Daniel.Guimarans@uab.cat

ABSTRACT

This article presents a Lagrangean Relaxation-based methodology to solve the Travelling Salesman Problem with Time Windows. The Lagrangean function exploits the structure of the problem. The proposed method, which elaborates on the Subgradient Optimization, presents a simple heuristics aimed to improve algorithm's convergence on the optimal solution. Furthermore, if the optimal solution is not found in a reasonable number of iterations, this method is able to provide a feasible solution while guaranteeing a tight gap between the primal and the optimal cost.

Keywords: Travelling Salesman Problem, Time Windows, Lagrangean Relaxation, Subgradient Optimization

1. INTRODUCTION

The Travelling Salesman Problem (TSP) is the following: "A salesman is required to visit once and only once each of n different cities starting from a base city, and returning to this city. What is the path that minimizes the total distance travelled by the salesman?" The TSP is probably the most famous and extensively studied problem in the field of Combinatorial Optimization (Lawer et al. 1985).

This work considers the Travelling Salesman Problem with Time Windows (TSP-TW) which is a generalization of the TSP. In the TSP-TW, the service at each city must start within an associated time window (Desrosiers et al. 1988).

The TSP belongs to the class of \mathcal{NP} -Hard optimization problems. Therefore, the TSP-TW with general time windows is at least \mathcal{NP} -hard too. Indeed, it is strongly \mathcal{NP} -complete to find a feasible solution for the TSP-TW (Savelsbergh 1985).

Lagrangean relaxation (LR) is a well-known method to solve large-scale combinatorial optimization problems. It works by moving hard to satisfy constraints into the objective function

associating a penalty on the objective if they are not satisfied. For a recent review of LR techniques and applications, see Guignard 2003.

In the proposed approach to the TSP-TW, LR relaxes the constraint set requiring that all customers must be served and the constraint set requiring that the arrival time must be greater than the earliest time.

The Lagrangean function exploits the structure of the problem, so it reduces considerably the formulation's size. Thus, the Lagrangean Problem needs less computational effort. However, it is often a major issue to find the optimal Lagrangean multipliers. The commonly used approach is the subgradient method. It guarantees convergence but it is too slow to be real practical interest.

The proposed method improves the convergence on the optimal solution in the Subgradient Optimization by using an Insertion heuristic to obtain a feasible solution from a LR solution. If the optimal solution is not found in a reasonable number of iterations, it is able to provide a feasible solution obtaining a tight gap between the primal and the optimal cost.

The present paper is structured as follows. Section 2 introduces some notation and terminology and presents a formulation of the problem. Section 3 describes the proposed Lagrangean relaxation. Section 4 describes the proposed method and heuristics used to improve its efficiency. Section 5 presents some results obtained by using the method proposed in the previous section. Finally, application and further research directions are outlined.

2. MATHEMATICAL FORMULATION

The TSP can be considered as a routing network, represented by a directed graph $G = (I, E)$, connecting city nodes $I = \{i_1, i_2, \dots, i_n\}$ through a set of directed edges $E = \{(i, j) : i, j \in I\}$, where $i_1 = 1$ is the base city. The tour is a path, $P = (1, i_2, \dots, i_n, 1)$

connected by edges belonging to E that starts and ends at a base city and visits all the cities.

An edge $e = (i, j) \in E$ is associated with a matrix $C = \{c_e\}$ denoting the travel time from node i to node j . It is assumed that distances defined in the problem satisfy the triangular inequality, that means the value c_e is supposed to be the lowest time route connecting node i to node j .

In the TSP-TW problem, each city $i \in I$ has an associated service time st_i and an associated time window $[a_i, b_i]$, defined by the earliest and the latest time to start the service in this city. The base city may also have a time window defining the scheduling horizon, but not a service time ($st_1 = 0$). Moreover there is a matrix (t_e) denoting the travelling time from node i to node j plus the service time st_i at node i , $t_e = st_i + c_e$.

Then a feasible solution to the TSP-TW should satisfy the following constraints:

- Each city should be visited only once.
- The route must start and finish at the base city.
- Each city is visited within its time window.

The problem goal is to minimize the total time of the route.

The proposed mathematical formulation requires defining the binary variable x_e to denote that the edge $e = (i, j) \in E$ is used in the tour

$$x_e = \begin{cases} 1 & \text{if city } j \text{ is visited immediately after } i; \\ 0 & \text{otherwise.} \end{cases}$$

It requires to define also another variable s_i to denote the arriving time at node i . These variables are nonnegative integers.

The proposed mathematical formulation for the TSP-TW problem is as follows:

$$\min \sum_{e \in E} t_e x_e \quad (1)$$

subject to

$$\sum_{e \in \delta^+(i)} x_e = 1, \quad \forall i \in I \quad (2)$$

$$\sum_{e \in \delta^-(i)} x_e = 1, \quad \forall i \in I \quad (3)$$

$$s_i + t_e - M(1 - x_e) \leq s_j, \quad \forall e = (i, j) \in E : j \neq 1 \quad (4)$$

$$s_i + t_e - M(1 - x_e) \leq b_1, \quad \forall e = (i, 1) \in E \quad (5)$$

$$s_i \geq a_i, \quad \forall i \in I \quad (6)$$

$$s_i \leq b_i, \quad \forall i \in I \quad (7)$$

where

- $\delta^+(i) = \{e \in E : \exists j \in I, e = (i, j)\}$ represents the set of arcs whose starting node is i .
- $\delta^-(i) = \{e \in E : \exists j \in I, e = (j, i)\}$ represents the set of arcs whose ending node is i .
- M is a large real value, satisfying $M \geq b_i + t_{ij} - a_j \quad \forall (i, j) \in E$.

Constraint (2) guarantees that every city must have one starting edge, whereas constraint (3) guarantees that every city must have one finishing edge.

If node i is visited immediately before node j , then constraint (4) states that the arriving time at node j , s_j , should be greater than s_i plus t_e , the sum of the travelling time and the service time at node i , except for the base city.

Constraint (5) states that the total time should be lower than the maximum allowed travelling time b_1 .

Constraints (6)-(7) impose that each city should be visited within the defined time window.

Note that constraint (4) replaces the subtour elimination constraints for the TSP. It was proposed by (Miller et al. 1960).

3. LAGRANGEAN RELAXATION

A feasible solution of a TSP is a 1-tree having two edges incident to each node, one starting edge and one ending edge. A 1-tree is a tree on the graph induced by $\{2, \dots, n\}$ nodes plus two edges incident to node 1, as explained on (Held and Karp 1970).

Note that summing constraints (2) and (3), the result is $\sum_{e \in E} x_e = n$. The 1-tree is determined by adding

to the latter expression the subtour elimination constraints. Finding a minimum weight 1-tree is relatively easy, so this is potentially an interesting relaxation.

However in the TSP-TW, the 1-tree should satisfy time windows constraints. Constraint (6) can increase the effort to find a minimum weight 1-tree.

The proposed Lagrangean function is as follows:

$$L(u, v, w) = \min_{x \text{ 1-tree}} \sum_{e \in E} t_e x_e + \sum_{i \in I} u_i \left(1 - \sum_{e \in \delta^+(i)} x_e \right) + \sum_{i \in I} v_i \left(1 - \sum_{e \in \delta^-(i)} x_e \right) + \sum_{i \in I} w_i (a_i - s_i) \quad (8)$$

$$\text{s.t. } s_i + t_e - M(1 - x_e) \leq s_j, \quad \forall e = (i, j) \in E : j \neq 1 \quad (9)$$

$$a_i + t_e - M(1 - x_e) \leq s_j, \quad \forall e = (i, j) \in E : j \neq 1 \quad (10)$$

$$s_i + t_e - M(1 - x_e) \leq b_1, \quad \forall e = (i, 1) \in E \quad (11)$$

$$a_i + t_e - M(1 - x_e) \leq b_1, \quad \forall e = (i, 1) \in E \quad (12)$$

$$s_i \leq b_i, \quad \forall i \in I \quad (13)$$

This LR relaxes constraints (2)-(3) weighting them with multiplier vectors u and v of appropriate dimensions and unrestricted sign. It also relaxes the constraint (6) weighting it with a multiplier vector w of appropriate dimensions and nonnegative components.

Note that, in this Lagrangean Dual Problem, waiting times are allowed. That means one may arrive at a city i earlier than a_i and wait until the node is released at time a_i . Thus, waiting time has influence on the dual cost of a solution. For this reason, the proposed primal formulation considers to minimize the total time of the tour.

As mentioned, finding a feasible solution for the TSP-TW is \mathcal{NP} -complete. However, this is a slightly relaxed problem and feasible solutions may be obtained more easily.

3.1. Strengthened Constraints

Since constraints (9), (11) and (13) are not very strong and they can be lifted in several ways, constraints (10) and (12) were added to avoid error propagation. Note that, constraints (9), (11) and (13) are the same as constraints (4), (5) and (7) respectively.

Also, to avoid the optimization to consider s_i bigger than the arrival time associated to x_e , Lagrangean function (8) is replaced by:

$$L(u, v, w) = \min_{x1-tree} \sum_{e \in E} t_e x_e + \sum_{i \in I} u_i \left(1 - \sum_{e \in \delta^+(i)} x_e \right) + \sum_{i \in I} v_i \left(1 - \sum_{e \in \delta^-(i)} x_e \right) + \sum_{i \in I} w_i \Delta_i \quad (14)$$

And constraints (15) and (16) are added:

$$\Delta_i \geq a_i - s_i \quad \forall i \in I \quad (15)$$

$$\Delta_i \geq 0 \quad \forall i \in I \quad (16)$$

3.2. Subgradient Optimization

Algorithm 1 shows the standard subgradient algorithm, as explained on (Wolsey 1998).

If $\lambda_k = \lambda_0 \rho^k$ for any parameter $\rho < 1$, then convergence is guaranteed if λ_0 and ρ are sufficiently large. Otherwise, geometric series $\lambda_0 \rho^k$ tend to zero too rapidly and sequences $\{u^k\}$, $\{v^k\}$ and $\{w^k\}$ converge before reaching an optimal point. In practice, rather than decreasing λ_k at each iteration, the parameter λ_k is reduced by ρ every n consecutive iterations without improving the lower bound.

Algorithm 1: Subgradient Optimization

Iteration 0
Initialize $u^0 \leftarrow v^0 \leftarrow w^0 \leftarrow 0$;
Iteration k
Solve the Lagrangean function $L(u^k, v^k, w^k)$
Evaluate the subgradient $\gamma(u^k, v^k, w^k)$
Calculate the step size λ_k
$(u^{k+1}, v^{k+1}, w^{k+1}) = (u^k, v^k, w^k) + \lambda_k \gamma(u^k, v^k, w^k)$
$k \leftarrow k + 1$

4. THE PROPOSED METHOD

Algorithm 2 shows the proposed method.

First, the method tries to obtain a feasible solution applying the Nearest-Feasible-Neighbour Heuristic. As mentioned above, it is important to notice that it is an \mathcal{NP} -complete problem to find a feasible solution to the TSP-TW. Therefore, if the heuristic only finds a subpath then it applies the Insertion Heuristic. Sorting visits dominates the time complexity of these heuristics.

Algorithm 2: The Proposed Method

Step 0
Initialize $u^0 \leftarrow v^0 \leftarrow w^0 \leftarrow 0$; $\delta_0 \leftarrow 2$
Apply the NFN Heuristic and the Insertion Heuristic.
Step 1
Solve the Lagrangean function $L(u^k, v^k, w^k)$
Evaluate the subgradient $\gamma(u^k, v^k)$
Step 2
if $(\ \gamma(u^k, v^k)\ ^2 = 0)$ then
It finds a tour perhaps it has a waiting time.
Step 3
if $(\ \gamma(u^k, v^k)\ ^2 < \zeta)$ then
Consider the subpath $P := (1, i_1, i_2, \dots, i_k, 1)$
from the dual solution and apply Insertion Heuristic
Step 4
Calculate the step size λ_k
$(u^{k+1}, v^{k+1}) = (u^k, v^k) + \lambda_k \gamma(u^k, v^k)$
$w_i^{k+1} = w_i^k + \lambda_k \max\{0, a_i - s_i\}$
$k \leftarrow k + 1$
Return to step 1.

In step 2, the method finishes if a tour is found. The dual solution may contain a waiting time, but the dual cost is the optimal cost. In this case, the solution is an equivalent of the optimal solution with some swapped nodes but identical cost. The waiting time can be eliminated afterwards by applying a Swap Heuristic or Two-Node-Exchange Heuristic, as explained on (Ascheuer et al. 2001).

In step 3, if the solution is nearly a tour, the method applies the Insertion Heuristic in the associated subpath. The parameter ζ depends on the number of variables. In the beginning, it may be large equals to $N/2$. However, it should be tight enough to not increase the running time excessively. Moreover, the method does not allow applying step 3 again in next consecutive iterations to not increase the running time significantly.

As mentioned above, the proposed method uses the step size λ_k , reducing by ρ every n consecutive iterations without improving the lower bound.

It should be remarked that the subgradient depends on constraint sets requiring that all customers must be served.

4.1. Nearest-Feasible-Neighbour Heuristic

Starting with each feasible edge $(1, i) \in E$, Nearest-Feasible-Neighbour Heuristic (NFN) enlarges the current subpath $(1, i_1, i_2, \dots, i_k)$ by an arc (i_k, i_l) resulting in the smallest increase in the objective value while guaranteeing feasibility, but the final path may not include all $j \in I$.

4.2. Insertion Heuristic

Starting with a shortest path in E obtained by the Nearest-Feasible-Neighbour Heuristic, Insertion Heuristic enlarges the current partial path $P := (1, i_1, i_2, \dots, i_k, 1)$ by a node j satisfying a certain insertion criterion. Let $S := I - \{i_1, \dots, i_k\}$ be the set of unvisited cities and

$$d_{\min}(j) := \min\{c_{i_l j} + c_{j i_{l+1}} - c_{i_l i_{l+1}} \mid i_l \in P \text{ and } (1, i_1, \dots, i_l, j, i_{l+1}, \dots, i_k, 1) \text{ is feasible}\}$$

Among all unsequenced cities, it chooses the node $j \in S$ that causes the lowest increase in the path length.

If the current partial path obtained from the dual solution has waiting time, the Insertion Heuristic enlarges the current partial path $P := (1, i_1, i_2, \dots, i_k, 1)$ in the first i_l which has $s_i > a_i$ by a node j satisfying a certain insertion criterion. Let $S := I - \{i_1, \dots, i_k\}$ be the set of unvisited cities and

$$d_{\min}(j) := \min\{c_{i_l j} + c_{j i_{l+1}} - c_{i_l i_{l+1}} \mid i_l \in P \text{ is the first waiting node, } a_j \leq s_j \leq b_j$$

$$\text{and } (1, i_1, \dots, i_{l-1}, j, i_l, \dots, i_k, 1), s_m \leq b_m \forall m \geq l\}$$

Among all unsequenced cities, it chooses the node $j \in S$ that causes the lowest increase in the path length. Of course, this reduces the waiting time in node i .

5. RESULTS

The method has been tested using the problems proposed by (Potvin and Bengio 1996) as individual route instances on Solomon's RC2 VRPTW (Vehicle Routing Problem with Time Windows). Solomon's

RC2 instances contain a combination of randomly-spaced and clustered customers. They have large time windows and large vehicle capacity but have very few routes and significantly more customers per route.

The Lagrangean function (14) has been programmed using the software ECLiPS_e and implemented using *eplex* library included in this package.

5.1. Heuristics Results

The heuristics were applied in the rc_201.1 individual route instance on Solomon's RC2 VRPTW instances.

In Fig. 1 the resulting path from using the Nearest-Feasible-Neighbour Heuristic (NFN) is shown. The path that was found is feasible but incomplete with $S = \{3, 16\}$.

In Fig 2. the resulting path using the Insertion Heuristic starting from the previous path is shown, but in this case the path found is complete but unfeasible.

In Fig. 3 the resulting path from using the Insertion Heuristic from the dual solution of $k = 13$ iteration is shown.

The results are shown on the table 1. Insertion Heuristic has completed the path from NFN Heuristic and obtained the optimal known travel time 592, but the path found has a waiting time. Also, the path found from the dual solution has obtained the optimal known travel time but with less travel cost.

Table 1: Heuristics' Results

Heuristic	Travel cost	Travel Time	Waiting Time
NFN	389	569	-
NFN + Insertion	366	592	36
Insertion	334	592	68

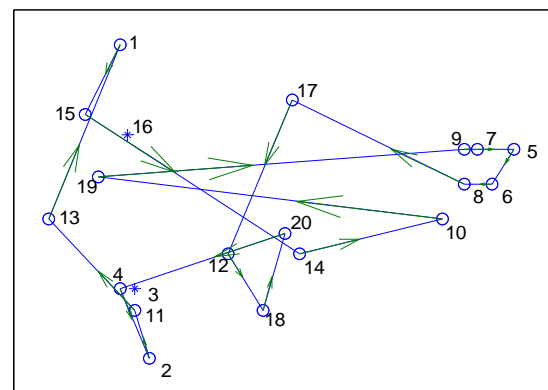


Figure 1: Nearest - Feasible - Neighbour Heuristic

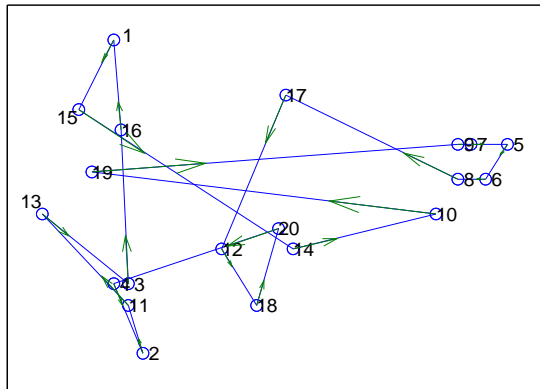


Figure 2: NFN Heuristic + Insertion Heuristic

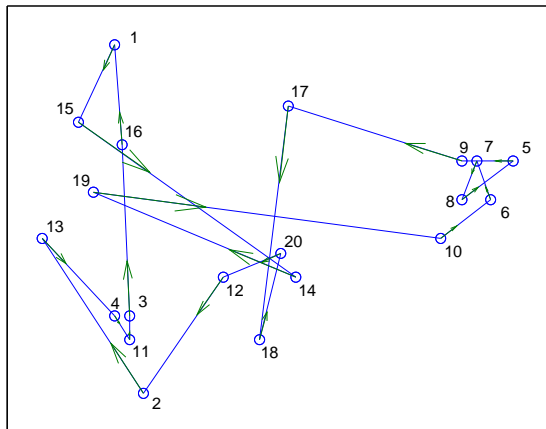


Figure 3: Insertion Heuristic

5.2. Method Results

The proposed method was applied in the individual route instances on rc_201 Solomon's RC2 VRPTW instance.

The results are shown on the table 2. Remember that the method minimizes the travel time, in that way the solutions found has obtained tight values in spite of the waiting time. But the travel cost has obtained a 27% deviation from the best known solution. The best known solution of the rc_201 Solomon's RC2 VRPTW is 1406.94 dist. with 4 vehicles, that was solved by Cordeau et. al. 2000.

Table 2: Method Results

Instances	Travel cost	Travel Time	Waiting Time
rc_201.1	366	592	36
rc_201.2	500	860	99
rc_201.3	521	862	31
rc_201.4	557	889	82
Total	1944	3203	248

Aiming to increase the proposed method's efficiency, the parameter $\lambda_k = \delta_k \frac{UB - L(u^k, v^k, w^k)}{\|\gamma(u^k, v^k, w^k)\|^2}$

with $0 < \delta_k \leq 2$ was introduced and UB denotes a known upper bound on the optimal value of the original problem. This step size guarantees convergence and leads to much faster convergence. However, depending on the problem instance, the method needs a too tight upper bound to ensure algorithm's convergence.

This step size was applied in the rc_201.1 individual route instance on Solomon's RC2 VRPTW instances. In Fig 4. the result using the UB equals to 592 is shown, in this case the sequences do not converge.

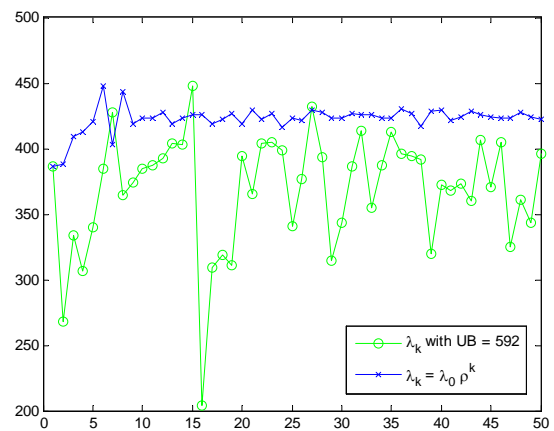


Figure 4: This step size does not converge

6. CONCLUSIONS AND FUTURE RESEARCH

The present paper presents an approximation to solve the Travelling Salesmen Problem with Time Windows using Lagrangean Relaxation. Moreover, the presented methodology, proposes an insertion heuristic to improve LR results.

The proposed method was found to be able to find feasible solutions of the problem allowing waiting time. However, the algorithm may not guarantee finding the optimal solution.

Since the future objective is to solve the VRPTW. Firstly, dividing it in instance of TSPTW clustering nodes in instances satisfy the constraint of vehicle capacity. Secondly, the proposed method will solve the instances. The solutions can be allow to have waiting times, even though, it may be not the optimal solution. Finally, the instances will be rescheduled according to the waiting time and time windows.

Near further research is to be addressed in the directions to improve convergence. Also introduce an algorithm to solve the Lagrangean function aiming to reduce the computational time.

REFERENCES

- Ascheuer, N., Fischetti, M., Grötschel, M., 2001. Solving the Asymmetric Travelling Salesman Problem with time windows by branch-and-cut. *Math. Program.*, Ser. A 90: 475-506.
- Cordeau, J. F., Laporte, G., Mercier, A., 2000. A unified tabu search heuristic for Vehicle routing problems with time Windows. Technical Report CRT-00-03, Centre for Research on Transportation, Montréal, Canada.
- Desrosiers, J., Sauvé, M., Soumis, F., 1988. Lagrangean relaxation methods for solving the minimum fleet size multiple travelling salesman problem with time windows. *Management Science* 34: 1005-1022.
- Guignard, M., 2003. Lagrangean Relaxation. *Top* 11(2): 151-228.
- Held, M., Karp, R. M., 1970. The traveling salesman problem and minimum spanning trees. *Operations Research* 18: 1138-1162.
- Lawer, E.L., Lenstra, J.K., Rinnoy Kan, A.H.G., Shmoys, D.B., 1985. *The Traveling Salesman problem: a guided tour of combinatorial optimization*. New York: John Wiley & Sons.
- Miller, C.E., Tucker, A.W., Zemlin, R.A., 1960. Integer programming formulation and travelling salesman problems. *J. Assoc. Comput. Mach.* 7: 326-329.
- Potvin, J. Y., Bengio, S., 1996. The Vehicle Routing Problem with Time Windows Part II: Genetic Search. *INFORMS Journal on Computing* 8: 165-172.
- Savelsbergh, M.W.P., 1985. Local search for routing problems with time windows. *Annals of Operations Research* 4: 285-305.
- Wolsey, L. A., 1998. *Integer programming*. New York : John Wiley & Sons.

AUTHORS BIOGRAPHY

Rosa Herrero is a PhD Student holding a fellowship at the Telecommunication and System Engineering Department at the Universitat Autònoma de Barcelona. She is a member of the Logisim Research Group. Her research interests are integer programming, lagrangean relaxation and simulation to solve industrial combinatorial problems.

Juan José Ramos is an Associated Professor at the Telecommunication and System Engineering Department at the Universitat Autònoma de Barcelona. He is coordinator and an active member of the Logisim Research Group. His research interests are modelling, simulation and optimisation of logistics problems.

Daniel Guimarans is a PhD Student and Assistant Teacher at the Telecommunication and System Engineering Department at the Universitat Autònoma de Barcelona. He is an active member of the Logisim Research Group. His research interests are constraint

programming, simulation and hybridisation of different techniques to solve industrial combinatorial problems.

COOPERATION AMONG PRODUCTION CHAINS TO MITIGATE THEIR ENVIRONMENTAL IMPACT

V. Albino^(a), A.C. Garavelli^(b), A. Messeni Petruzzelli^(c), D.M. Yazan^(d)

DIMeG, Politecnico di Bari, Viale Japigia 182 – 70126, Bari, Italy

^(a) v.albino@poliba.it, ^(b) c.garavelli@poliba.it, ^(c) a.messeni.petruzzelli@poliba.it, ^(d) d.yazan@poliba.it

ABSTRACT

Enterprises are exploring new collaborative relationships aimed at reducing the environmental impact of their production processes and chain. Different cases of collaboration motivated by environmental goals are emerging and some of them are based on the industrial symbiosis.

In this paper, production chains aimed at mitigating their environmental impact adopting industrial symbiosis approach are considered. Using enterprise input-output models, different types of cooperation among production chains are identified and classified based on materials and energy flows exchanges. In particular, we identify some environmental, economic, and operational conditions that have to be respected in order to permit cooperative relationships. A case example related to the concrete and marble industries is analyzed and discussed.

Keywords: production chains, cooperation, industrial symbiosis, enterprise input-output

1. INTRODUCTION

Environmental sustainability is becoming the reference issue of any development program since the attention towards the continued deterioration of the global environment and the unsustainable pattern of consumption and production is increasing (EU 2008; OECD 2008; UNEP 2008). Improving efficiency in production processes, and sustainability in the use of resources permit the shift towards sustainable consumption and production. In particular, the Kyoto protocol is forcing a significant set of country economies to respect some environmental constraints. Consequently, in such economies production chains and then their firms have to contribute to this goal. At the present, there is a need to give further impetus to efficient and eco-innovative production processes, to reduce dependency on raw materials, and to encourage optimal resource use and recycling (UNEP 2009).

Some integrated actions can be proposed such as: i) boosting resource efficiency; ii) supporting eco-innovation; and iii) enhancing the environmental potential of industry. Resource efficiency contributes to the goal of creating more value while using less resources. More tools have to be developed to monitor, benchmark and promote resource efficiency, taking into account a life-cycle perspective. Detailed material-based analysis and targets have also to be addressed, focusing on environmental significance and on access to natural resources.

To develop resource-efficient and eco-innovative production processes, to reduce dependency on raw materials and to encourage optimal resource use and recycling different strategies are possible for industrial firms (Korhonen 2004). In particular, dematerialization strategies aimed at reducing material flows and their impacts, play an important role in the sustainable production policies (Nathani 2000). Beside the development of new technologies for dematerialization, other methods such as recycling and alternative materials use can be effective. Often, this is possible if cooperation between firms of a production chain or of different production chains arises. Such a cooperation aimed at mitigating the environmental impact of production chains mostly refers to the industrial ecology which is based on the idea that industrial eco-systems may behave similar to the natural eco-systems. Eco-industrial parks (Cote and Cohen-Rosenthal 1998) and industrial symbiosis (Chertow 2000) represent different forms of cooperation between actors of production chains.

The cooperation among the actors of a production chain can support the improvement of environmental performance of each actor as well as of the whole production chain. For instance, the recycle of wastes of an actor as primary inputs for another actor can result in a form of industrial symbiosis with a double benefit: the reduction of waste disposal and of primary inputs purchase. Both environmental impacts and related costs can then be mitigated if they cooperate. Moreover, cooperation may involve actors of different production chains. In all cases, the mutual benefit as well as the cost of cooperation have to be considered. The share of the net benefit among the actors of the chains represents the incentive to explore the space of cooperation.

In this paper we propose to represent a production chain, whose wastes or by-products can be used as inputs of another chain, adopting an Enterprise Input-Output (EIO) approach (Grubbström and Tang 2000; Polenske 2001). Such an approach has been used to model complex supply chains (Albino, Izzo, and Kultz 2002) to evaluate the effect of different coordination policies of materials flows on the logistics and environmental performance of an industrial district (Albino, Kultz, Messeni Petruzzelli 2008), or to minimize the waste and resource consumption of a manufacturing system (Zue, Kumar, and Sutherland 2007).

In the paper, an EIO model of two independent chains is developed to evaluate the cooperation opportunity in terms of both economic and environmental benefits. Operational aspects, such as the

logistic one, are also considered since they may strongly affect the cooperative relationship and the rise of benefits. The model will be applied to a case example where a marble tiles production chain is linked with a concrete production chain throughout the use of marble powder as an alternative aggregate instead of traditional aggregates.

2. COOPERATION FOR SUSTAINABLE PRODUCTION CHAINS

From a physical point of view, a production chain can be considered as an input-output system (Storper and Harrison 1998) that describes the product flows existing among production processes. Input-output systems may involve many production processes, depending on a specific division and classification of production in processes. In general, the chain may contain the extraction of raw materials, manufacturing, distribution and use of goods. Referring to the environmental sustainability, a sustainable production chain means favouring, for a given output, processes with less use of natural resources (energy and materials) and production of wastes. Regarding the chain as a whole, different terms have been proposed by the literature to define the environmental behavior of the production chain. The concept of the environmental chain (Bloemhof-Ruwaard, van Beek, Hordijk, Van Wassenhove 1995) has been introduced to focus on the evaluation of the emissions caused by the supply chain in the environment. Differently, the concept of “extended” or “green” supply chain arises as the basic structure of the entire supply chain needs re-definition by accommodating environmental concerns of waste and resource use minimisation (Beamon 1999). This means that the traditional structure of supply chain must be extended to include mechanisms for materials and energy recovery. In particular, extended supply chains include the reduction and elimination of by-products through cleaner process technologies and lean production techniques. Moreover, this concept raises from the extension of supply chain boundaries as far as to include the source and the destination of all the physical flows used and produced at each supply chain stage. Spatial aspects can also be relevant to evaluate the environmental impact of production chains since environmental and economic performance measures are strongly related with transportation and logistics solutions (Albino, Izzo, and Kuitz 2002; Albino, Kuitz, Messeni Petruzzelli 2008). Then, increasing attention is now paid to operations management in sustainable supply chains (Linton, Klassen, Jayaraman 2007). As the opportunity offered by the efficient use of materials (recycling, etc.) and energy stresses the relevance of network of actors, cooperation between actors becomes crucial.

Cooperation can be considered also in terms of industrial ecology. Scholars (Frosch and Gallopoulos 1989) introduce the concept of “industrial ecosystem” as a new way to integrate industrial processes that transform raw materials into final products and waste to be disposed of. In fact, the consumption of energy and materials needs to be optimized, waste production reduced, and the outputs of one process have to serve as inputs for other processes. In particular, (Gertler 1995) defines an industrial ecosystem as a community or

network of companies in a region who interact by exchanging and making use of by-products and/or energy in a way that provides some benefits, such as increased energy efficiency reducing systemic energy use, and reduced waste products requiring disposal. Then, co-location permits the materials/energy exchanging networks to work better (Lowe 1997). However, the most critical innovation of the industrial ecology concept is the level of inter-enterprise cooperation (Wallner 1999). In fact, it is not the single component of the production system, the firm, that is the subject of analysis, but the network of region-wide settled enterprises for which a spatial proximity does exist. Moreover, since many problems have their roots in local activities, the participation and cooperation of local authorities may be a determining factor for the success (Zuindeau 2006). A broad variety of networks exists. Industrial symbiosis is the main approach introduced by the industrial ecology to explain how industrial ecosystem may work. Two industrial actors operate in industrial symbiosis when they exchange materials and energy to reduce costs, create value, and improve the environment. They share by-products, services, and information in a cooperative and synergistic way which is strongly supported by the geographical proximity. Eco-industrial parks represent a set of businesses and a local community aiming at reducing their environmental impact and gaining economic and social benefits. This approach can be more extended than the industrial symbiosis covering all activities not strictly related to the exchange of materials and energy such as logistics and shipping, green building retrofit, education (Cote and Cohen-Rosenthal 1998). In general, industrial symbiosis as well as eco-industrial park may be the result of a bottom-up initiative, as the environmental regulations become stricter, or of a top-down initiative based on planning, design, and construction phases. Both cases are described in the literature and quantitative approaches have been proposed to support sustainability evaluation. However, the analysis of the role and level of cooperation between actors of production chains in permitting environmental and economic benefits needs to be explored further.

To this aim, in this paper the enterprise input-output approach is proposed to build quantitative models of materials and energy exchange which include cooperation analysis.

3. THE MODEL

We consider the EIO model for a production chain. Let Z_0 be the matrix of domestic (i.e. to and from production processes belonging to the production chain) intermediate deliveries, f_0 is the vector of final demands, and x_0 the vector of gross outputs. If n processes are considered the matrix Z_0 is of size $n \times n$, and the vectors f_0 and x_0 are $n \times 1$. It is assumed that each process has a single main product as its output. Each of these processes may require intermediate inputs from the other processes, but not from itself so that the entries on the main diagonal of the matrix Z_0 are zero. Of course, also other inputs are required for the production. These are s primary inputs (i.e. products not

produced by one of the n production processes). Next to the output of the main product, the processes also produce m by-products and waste. r_0 and w_0 are the primary input vector, and the by-product and waste vector of size $s \times 1$ and $m \times 1$, respectively.

Define the intermediate coefficient matrix A as follows:

$$A = Z_0 \hat{x}_0^{-1} \quad (1)$$

where a "hat" is used to denote a diagonal matrix. We now have:

$$x_0 = Ax_0 + f_0 = (I - A)^{-1} f_0 \quad (2)$$

It is possible to estimate R , the $s \times n$ matrix of primary input coefficients with element r_{kj} denoting the use of primary input k ($1, \dots, s$) per unit of output of process j , and W , the $m \times n$ matrix of its output coefficients with element w_{kj} denoting the output of by-product or waste type l ($1, \dots, m$) per unit of output of product j . It results:

$$r_0 = R x_0; w_0 = W x_0 \quad (3)$$

Note that the coefficient matrices A , R , and W are numerically obtained from observed data and calculated under the assumption of fixed technology.

Input-output techniques may be adopted to develop and model efficient policies aimed at a more efficient and sustainable use of primary resources and wastes management. Specifically, such policies may be applied at the level of a single production chain as well as of a set of production chains which aim at establishing an industrial symbiosis.

Let us consider two independent production chains, α and β , whose input-output equations are formulated as:

$$x_0^\alpha = (I - A^\alpha)^{-1} f_0^\alpha \quad (4)$$

$$r_0^\alpha = R^\alpha x_0^\alpha; w_0^\alpha = W^\alpha x_0^\alpha \quad (5)$$

$$x_0^\beta = (I - A^\beta)^{-1} f_0^\beta \quad (6)$$

$$r_0^\beta = R^\beta x_0^\beta; w_0^\beta = W^\beta x_0^\beta \quad (7)$$

The potential environmental and economic opportunities rising from an industrial symbiosis between α and β chains can be investigated. In particular, such opportunities may involve the abatement or reduction of by-products and wastes disposal costs, and/or the saving of primary inputs such as material resources. For instance, one specific waste of the production chain α (or β) can be recycled or used as primary input of the production chain β (or α).

Let us assume that chain α produces wastes that may be used without any further transformation as primary inputs by chain β . In particular, for the sake of simplicity, let us assume that both α and β are constituted

by a single production process, and that the former produces only one waste (w^α), that is disposed of to the landfill l , and the latter utilizes only one primary input (r^β), that is purchased by the seller s .

Let us consider two different scenarios. In particular, in the first one, scenario A, the production chains do not cooperate, whereas in the second scenario, scenario B, they cooperate. In the scenario A, the costs associated to the disposal of w^α ($C_{w^\alpha}^A$) and the purchase of r^β ($C_{r^\beta}^A$) can be estimated as, respectively:

$$C_{w^\alpha}^A = C_{w^\alpha}^l \cdot w^\alpha + C_{w^\alpha}^{ad} \cdot w^\alpha \quad (8)$$

$$C_{r^\beta}^A = C_{r^\beta}^s \cdot r^\beta + C_{r^\beta}^{s\beta} \cdot r^\beta \quad (9)$$

where $C_{w^\alpha}^l$ and $C_{r^\beta}^s$ represents the unitary cost to dispose of w^α and to purchase r^β , respectively, and $C_{w^\alpha}^{ad}$ and $C_{r^\beta}^{s\beta}$ represents the unitary cost to transport w^α from α to l , and r^β from s to β . Thus, the total cost of disposal and purchase associated to the scenario A can be calculated as:

$$C^A = C_{w^\alpha}^A + C_{r^\beta}^A \quad (10)$$

According to the different quantity of w^α and r^β produced and required, respectively, three situations may occur, i.e.: 1) $w^\alpha > r^\beta$, 2) $w^\alpha = r^\beta$, and 3) $w^\alpha < r^\beta$.

Let us consider the first situation, $w^\alpha > r^\beta$. Therefore, in the scenario B, since the production chains cooperate, the cost associated to the use of w^α for substituting r^β , $C_{w^\alpha r^\beta}^B$, and to the disposal of the remaining amount $w^\alpha - r^\beta$, $C_{w^\alpha}^B$, can be estimated as, respectively:

$$C_{w^\alpha r^\beta}^B = C_{w^\alpha r^\beta}^{\alpha\beta} \cdot r^\beta \quad (11)$$

$$C_{w^\alpha}^B = C_{w^\alpha}^l \cdot (w^\alpha - r^\beta) + C_{w^\alpha}^{ad} \cdot (w^\alpha - r^\beta) \quad (12)$$

where $C_{w^\alpha r^\beta}^{\alpha\beta}$ represents the unitary cost to transport the required quantity r^β from α to β .

Both the chains have to identify which has to support this actual cost as well as the eventual exchange cost (and revenue) related to the use of the waste as primary input. However, since this exchange cost (and revenue) does not affect the evaluation of the cooperation opportunity, it may be neglected. Thus, the total cost associated to the scenario B can be calculated as:

$$C^B = C_{w^\alpha r^\beta}^B + C_{w^\alpha}^B \quad (13)$$

Let us consider $\Delta_{AB} = C^A - C^B$. If $\Delta_{AB} > 0$, the two production chains may fruitfully cooperate, since it

determines an economic benefit. Assuming that the cooperation is fruitful it is possible to analyse how w^α affects the total costs associated to the scenarios A and B, and, consequently, the economic benefit (Δ_{AB}). In Figure 1, the relationships linking C^A and C^B with the ratio between w^α and r^β are presented, assuming r^β as constant. Specifically, it is possible to notice that C^A is positively related to w^α , whereas C^B decreases up to point H, where w^α is equal to r^β , and then it increases with the same slope of C^A .

Figure 1 permits also to evaluate the economic benefit as the difference between C^A and C^B . In particular, when w^α is equal to r^β the economic benefit, in terms of cost savings, reaches its maximum value, since the cost is represented only by $C^B_{w^\alpha r^\beta}$. In addition, in this point also the environmental benefit is the greatest one, since no waste has to be disposed of and no primary input has to be purchased. Thus, the internalization of positive externalities should also be accounted to properly evaluate Δ_{AB} . The production chain having the greater economic benefit, as cost savings, it is also the one having more willingness to cooperate.

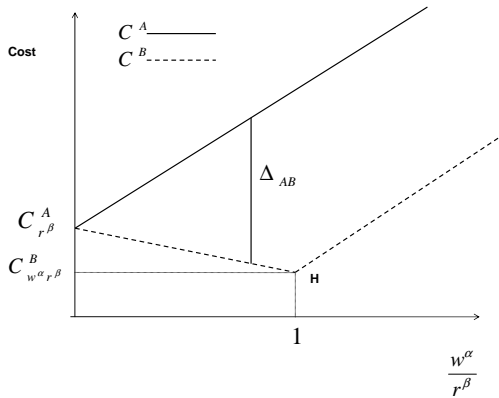


Figure 1: Cooperation economic benefit with r^β as constant.

According to the different value of $\frac{w^\alpha}{r^\beta}$, we can model the production chains and their link by different input-output equations. Specifically, three situations may occur: 1) $w^\alpha = r^\beta$, 2) $w^\alpha > r^\beta$, and 3) $w^\alpha < r^\beta$. In the first situation, we have:

$$x^\alpha = f^\alpha; x^\beta = f^\beta; W^\alpha x^\alpha = R^\beta x^\beta \quad (14)$$

and, consequently, it is possible to evaluate the relationship between the two final demands as $\frac{f^\beta}{f^\alpha} = \frac{W^\alpha}{R^\beta}$. This equation characterizes the two production chains in condition of perfect symbiosis. In the second situations, the equations can be formulated as:

$$x^\alpha = f^\alpha; x^\beta = f^\beta \quad (15)$$

$$w^{*\alpha} = W^\alpha x^\alpha - R^\beta x^\beta = W^\alpha f^\alpha - R^\beta f^\beta \quad (16)$$

where $w^{*\alpha}$ represents the residual amount of waste that has to be disposed of to the landfill. Finally, if $w^\alpha < r^\beta$, it results:

$$x^\alpha = f^\alpha; x^\beta = f^\beta \quad (17)$$

$$r^{*\beta} = R^\beta x^\beta - W^\alpha x^\alpha = R^\beta f^\beta - W^\alpha f^\alpha \quad (18)$$

where $r^{*\beta}$ represents the residual amount of primary input that has to be purchased. Following these three different situations, we can model the relationship between f^α and f^β as shown in Figure 2, where OA depicts the relationship when the perfect symbiosis occurs. Differently, if $w^\alpha > r^\beta$, the relationship between f^α and f^β occurs in the area below OA, and more chains β have to be involved in order to achieve the perfect symbiosis. Similarly, if $w^\alpha < r^\beta$, the relationship between f^α and f^β occurs in the area above OA, and more chains α have to be involved in order to achieve the perfect symbiosis.

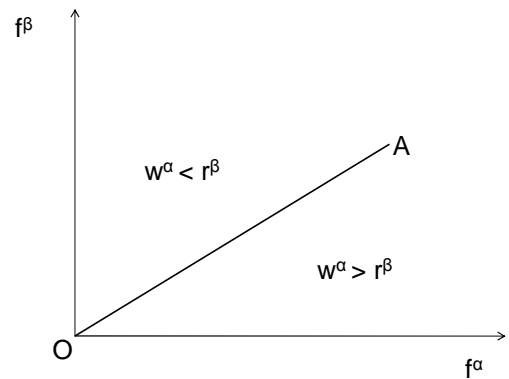


Figure 2: Symbiosis relationship between production chains' final demands.

A more complex situation may also be investigated, where also chain α uses the waste of β as primary input resulting in a roundput system [5]. In this case, if both $w^\alpha = r^\beta$ and $w^\beta = r^\alpha$, the input-output equations can be formulated as:

$$x^\alpha = f^\alpha; x^\beta = f^\beta \quad (19)$$

$$W^\alpha x^\alpha = R^\beta x^\beta; W^\beta x^\beta = R^\alpha x^\alpha \quad (20)$$

$$\text{with: } \frac{f^\beta}{f^\alpha} = \frac{W^\alpha}{R^\beta} \text{ and } \frac{f^\beta}{f^\alpha} = \frac{R^\alpha}{W^\beta}$$

Therefore, the relationships between f^α and f^β can be described as represented in Figure 3, where OA

depicts the relationship when the perfect symbiosis occurs between w^α and r^β , whereas OB depicts the relationship when the perfect symbiosis occurs between w^β and r^α . In this case, as clearly shown in Figure 3, the whole perfect symbiosis only occurs when $\frac{R^\alpha}{W^\beta} = \frac{W^\alpha}{R^\beta}$.

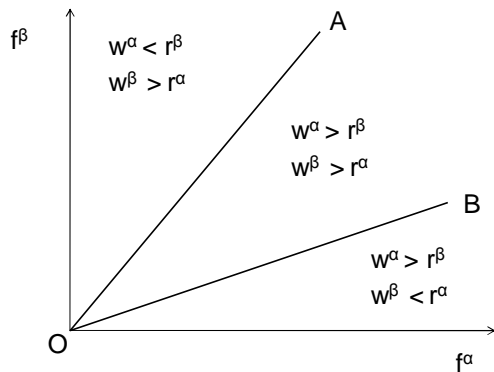


Figure 3: Symbiosis relationship between production chains' final demands in the case of roundput system.

In the case of roundput system, the cooperation economic benefit may be evaluated (Figure 4). Specifically, C^{B1} and C^{B2} represents the cost associated to the cooperating scenarios when w^α is used to substitute r^β , and w^β is used to substitute r^α , respectively; C^B represents the cost associated to the whole cooperating scenario, i.e. the sum of C^{B1} and C^{B2} .

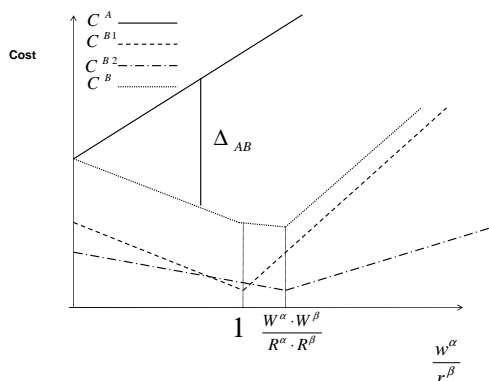


Figure 4: Cooperation economic benefit in the case of roundput system with r^β as constant.

4. CASE EXAMPLE

In the present section an actual example related to the cooperation between a concrete production chain and a marble production chain is presented adopting the proposed enterprise input-output approach. Nowadays, concrete chain producers are developing sustainable strategies to reduce environmental, social, and economic impacts of this industry, as well shown by "The cement sustainability initiative" signed in 2002 by the ten biggest cement producers of the world (WBCSD, 2002). One of the mentioned strategies is based on the use of wastes and by-products deriving from other industries (e.g. wastes of pneumatics, inert materials deriving from construction and demolition processes, marble industry

residuals) as alternative aggregates in concrete production. The use of these wastes and by-products helps not only to reduce traditional aggregate extraction but also to reduce the disposal impact of these alternative aggregates (Rao, Jha, and Misra, 2007). In this case, referring to an Italian geographic area specialised in the production of marble, two distinct production chains represented by a marble production (α) and by a concrete production (β) have been considered. Both chains are modelled as a single process chain having chain α a single waste and chain β a single primary input (see Table 1).

Table 1. Processes, main products, wastes and primary inputs of the chains.

	Process	Main product	Primary input	Waste
Chain α	Marble production	Marble [ton]	-	Marble residuals [ton]
Chain β	Concrete production	Concrete [ton]	Coarse aggregate [ton]	-

The chain α produces, as a waste, marble residuals (w^α), that may be used by the chain β to substitute the primary input of traditional coarse aggregates (r^β). If the two chains don't cooperate, w^α is disposed of to the landfill l and r^β is purchased by the seller s . In Table 2, the total amount w^α annually produced by α and of r^β annually required by β are presented.

Table 2. Annual amount w^α and r^β .

	Annual amount
w^α	500 ton/year
r^β	1000 ton/year

In Table 3, the unitary costs to dispose of w^α and to purchase r^β are shown.

Table 3. Unitary costs to dispose of w^α and to purchase r^β .

	Unitary cost
$C_{w^\alpha}^l$	20 €/ton
$C_{r^\beta}^s$	100 €/ton
$C_{w^\alpha}^{dl}$	40 €/ton
$C_{r^\beta}^{s\beta}$	80 €/ton

Assuming that the two production chains do not cooperate (scenario A), the total cost can be computed as reported in equation 10. Therefore, C^A is equal to 210.000 €/year. If the two production chains cooperate in order to reduce cost and to mitigate the environmental

impact, the overall cost structure associated to this different scenario (B) changes. Specifically, setting $C_{w^{\alpha}r^{\beta}}^{\alpha\beta}$ equals to 50 €/ton, we have:

$$C_{w^{\alpha}r^{\beta}}^B = C_{w^{\alpha}r^{\beta}}^{\alpha\beta} \cdot w^{\alpha} = 25.000 \text{ €/year}$$

$$C_{r^{\beta}}^B = C_{r^{\beta}}^s \cdot (r^{\beta} - w^{\alpha}) + C_{r^{\beta}}^{s\beta} \cdot (r^{\beta} - w^{\alpha}) = 90.000 \text{ €/year}$$

Thus, the total cost C^B is equal to 115.000 €/year and the overall economic benefit (cost savings) due to the production chains cooperation (Δ_{AB}) is equal to 95.000 €/year, as shown in Figure 5.

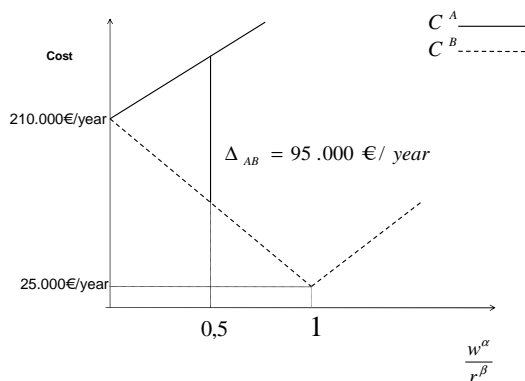


Figure 5: Cooperation economic benefit with r^{β} as constant.

Taking into account the actual cost structure it is also possible to identify which chain has more willingness to cooperate. In particular, in the case of cooperation, the chain α has an economic benefit equal to:

$$C_{w^{\alpha}}^l \cdot w^{\alpha} + C_{w^{\alpha}}^{ol} \cdot w^{\alpha} = 30.000 \text{ €/year}$$

since no wastes have to be disposed of to the landfill l . On the other hand, the chain β has an economic benefit equal to:

$$C_{r^{\beta}}^s \cdot r^{\beta} + C_{r^{\beta}}^{s\beta} \cdot r^{\beta} - C_{r^{\beta}}^s \cdot (r^{\beta} - w^{\alpha}) + C_{r^{\beta}}^{s\beta} \cdot (r^{\beta} - w^{\alpha}) = 90.000 \text{ €/year}$$

Thus, the chain β is more favouring to cooperate, since it may achieve the greater benefit.

In addition, it is possible to evaluate the relationship between f^{α} and f^{β} . In particular, since $\frac{w^{\alpha}}{r^{\beta}}$ is equal to 0,5, the relationship occurs in the area above OA (Figure 2), and, thus, more marble production chains have to be involved in order to achieve the perfect symbiosis.

5. CONCLUSIONS

In this paper an EIO model is proposed to evaluate the opportunity of cooperation between production chains aimed at reducing their environmental impact. The use of wastes of a chain as primary input of the other one has been considered as an industrial symbiosis of two

simple chains (one process each, one waste and one primary input without recycling). Two scenarios have been compared: no-cooperation vs. cooperation. Results show that the economic benefit of cooperation depends on the cost of disposal of waste, of purchase of primary input, and of transportation. This benefit will be shared between the chains depending also on the different willingness to cooperate. This condition has been related to the corresponding cost savings. The environmental benefit is also identified in terms of reduced primary input need and waste disposal. The perfect symbiosis has been defined referring to the enterprise input-output model in terms of final demand ratio resulting in a perfect balance between wastes produced and primary input needed. A more complex situation has been considered when the waste of each chain is used as primary input of the other one (roundput system). Perfect symbiosis conditions usually differ for the two flows. The cooperation benefit has been evaluated for this case. In the case example, the cooperation opportunity of marble and concrete production chains has been investigated. Economic and environmental benefits for the specific cooperation have been evaluated.

Further research can regard more complex links between two chains or among more chains, the effect of recycling, and the management of logistics if many chains are considered. Moreover, how the economic benefit is shared between firms is another important issue to be more in-depth analyzed.

REFERENCES

- Albino, V., Izzo, C., Kuhtz, S., 2002. Input-output models for the analysis of a local/global supply chain, *International Journal of Production Economics*, Vol.78, pp.119-131.
- Albino, V., Kuhtz, S. Messeni Petruzzelli, A., 2008. Analysing logistics flows in industrial clusters using an enterprise input-output model, *Interdisciplinary Information Sciences*, Vol.14, No.1, pp.1-17.
- Beamon, B. M., 1999. Designing the Green Supply Chain, *Logistics Information Management*, Vol.12, pp.332-342.
- Bloemhof-Ruwaard J.M., van Beek P., Hordijk L., Van Wassenhove L.N., 1995. Interactions between operational research and environmental management, *European Journal of Operational Research*, Vol.85, pp.229-243.
- Chertow M.R., 2000. Industrial symbiosis: literature and taxonomy, *Annual Review of Energy Environment*, Vol.25, pp.313-337.
- Cote P., Cohen-Rosenthal E., 1998. Designing eco-industrial parks: a synthesis of some experience, *Journal of Cleaner Production*, Vol.6, pp.181-188.
- European Union, 2008. European Sustainable Consumption and Production Policies, ec.europa.eu/environment/eussd/escp_en.htm.
- Frosch D., Gallopoulos N., 1989. Strategies for manufacturing, *Scientific American*, Vol. 261, No. 3, pp.94-102.
- Gertler N., 1995. Industrial ecosystems: developing sustainable industrial structures, *Dissertation for Master of Science in Technology and Policy and Master of*

Science in Civil and Environmental Engineering, Massachusetts Institute of Technology, Cambridge, MA.

Grubbström, R.W., Tang, O., 2000. An overview of input-output analysis applied to production-inventory systems, *Economic Systems Research*, Vol. 12, pp.3-25.

Korhonen J., 2004. Industrial ecology in the strategic sustainable development model: strategic applications of industrial ecology, *Journal of Cleaner Production*, Vol.12, pp.809-823.

Linton J.D., Klassen R., Jayaraman V., 2007. Sustainable supply chains: An introduction, *Journal of Operations Management*, Vol.25, No.6, pp.1075-1082.

Lowe E.A., 1997. Creating by-product resource exchanges: strategies for eco-industrial parks, *Journal of Cleaner Production*, Vol.5, No.1-2, pp.57-65.

Nathani, C., 2000. Linking an IO model with a material flow model to assess structural change related to material efficiency strategies. An application to the paper cycle, *Proceedings of the 13th International Input-Output Conference*, Macerata, Italy, 21-25 August.

OECD, 2008. *Measuring Sustainable Production*, OECD, Paris.

Polenske, K.R., 2001. Competitive advantage of regional internal and external supply chain, in Lahr, M., Miller, R. (Eds.), *Perspective in regional science: a Festschrift in Memory of Benjamin Stevens*. Elsevier, Amsterdam.

Rao, A., N. Jha, K., Misra, S., 2007. Use of aggregates from recycled construction and demolition waste in concrete, *Resources Conservation & Recycling*, Vol. 50, pp.71-81.

Storper, M., Harrison, B., 1992. Flessibilità, gerarchie e sviluppo regionale: la ristrutturazione organizzativa dei sistemi produttivi e le nuove forme di governance, in Belussi, F. (ed.), *Nuovi modelli d'impresa, gerarchie organizzative e impresa rete*, Angeli, Milano, pp.209-237.

United Nations Environment Programme (UNEP), 2009. *Global Green New Deal. Policy Brief*, www.unep.org/pdf/GGND_Final_Report.pdf.

United Nations Environment Programme (UNEP), 2009. *UNEP Year Book 2009*, www.unep.org/publications/UNEP-eBooks/UNEP_YearBook2009_ebook.pdf.

Wallner H.P., 1999. Towards sustainable development of industry: networking, complexity and eco-clusters, *Journal of Cleaner Production*, Vol.7, pp.49-58.

World Business Council for Sustainable Development (WBCSD), 2002. *Cement Sustainability Initiative*. WBCSD, Geneva.

Zue, H., Kumar, V., Sutherland J.W., 2007. Material flows and environmental impacts of manufacturing systems via aggregated input-output models, *Journal of Cleaner Production*, Vol. 15, pp.1349- pp.1349-1358

Zuindeau B., 2006. Spatial Approach to Sustainable Development: Challenges of Equity and Efficacy, *Regional Studies*, Vol.45, pp.459-470.

AUTHORS BIOGRAPHY

Vito Albino got the laurea degree in Mechanical Engineering. Visiting scholar (1986) at the University of Cincinnati (USA) and visiting professor (1994) at the

University of South Florida (Tampa, USA), from 1988 to 2000 he has taught Engineering Management at the University of Basilicata, Italy. He is now Full Professor of Innovation and Project Management at the Polytechnic of Bari, Italy. Member of several national and international associations, he is author of more than 70 papers published on national and international journals and conference proceedings in the fields of project management, operations management, firm clusters.

A. Claudio Garavelli, Ph.D. in Engineering Management, he has been Assistant Professor at the University of Basilicata,

Italy, since 1994. Visiting scholar in 1996 at the University of South Florida (Tampa, USA), he was formerly Associate Professor at the University of Lecce, then at the Polytechnic University of Bari, where he is at present Full Professor. His main teaching and research areas concern operations management, decision support systems, organisation clusters. He is involved in many national and international research projects and he is author of more than 80 papers published on national and international journals and conference proceedings.

Antonio Messeni Petruzzelli got the laurea degree in Business Engineering at the Politecnico di Bari. After a two years period as organizational analyst at Eni SpA, he got the Ph.D. in Management at same university. Visiting scholar at the IESE Business School, his research interests mainly concern the area of innovation management, including themes such as knowledge creation and transfer, university-industry relationships, proximity, system dynamics modelling, and input-output analysis. In these topics he has published several articles on international journals and presented paper at international conferences.

Devrim M. Yazan is a 3rd year Ph.D. student at Scuola Interpolitecnica a joint PhD program of the three Italian Technical Universities: Bari, Milan and Turin. He got Luarea degree at Politecnico di Bari. His main research project concerns the input-output analysis and the supply chain management.

DEMAND FORECASTING. THE CASE OF AN INTERNATIONAL FURNITURE COMPANY

Stefano SAETTA^(a), Lorenzo TIACCI^(b)

Dipartimento di Ingegneria Industriale
Università di Perugia

^(a) stefano.sietta@unipg.it, ^(b) lorenzo.tiacci@unipg.it

ABSTRACT

In a highly competitive market, characterized by increasing risk of obsolescence and short response time, companies are increasingly urged to adopt systems for forecasting demand which would allow them to determine the most appropriate strategies. The purpose of the paper is to identify an optimal solution to the forecasting of demand for improved cost management and the achievement of an efficient and responsive management for the case of office furniture company. Forecasts on future sales have a significant operational importance: in the short term, to organize the resources, the business and strategic functions, in long term to decide the investment programs. Demand Forecasting is the first step in the management of a modern Supply Chain. In the paper a case of demand forecasting improvement is shown. Considered company former forecasting method is based on the moving average, on a monthly scale, to make estimation for the following period. Paper goal is to find, if possible, a new model to increase the accuracy of the estimation. Classical forecasting method were used.

Keywords: demand forecasting, case study, demand in office furniture, exponential average, Holt model.

1. INTRODUCTION

Demand forecasting, i.e. the studies of the best method for reading the future number of sales, even if is an hard or even impossible task, is always appealing for researchers and management.

It is possible to say that at least 4 fields have been studied in the last years:

1. Forecasting energy demand;
2. Forecasting tourism demand;
3. Forecasting intermittent demand;
4. Forecasting integration in supply chain management.

Considering the first field, in the last years energy has been an issue. In every countries, but especially in the countries with a rapid growing GDP, energy consumption has been raising a lot. Also in countries like Europe there is a big concern about energy,

especially because of the risk to be too much dependent on energy supplier countries. In this sense it became the more and more important to be able to forecast future needs. In (Han 2009) the problem of electric load forecast for decision makers is considered. He gives an overview of the most used forecasting methods. In (Day 2009), authorities want to know the impact of the need of codlings in term of CO₂ emissions up to 2030 for a big city like London. The author describes a model for analyzing the cooling requirements in key buildings. It evaluates also how requirements could be affected by change in system mix, improved efficiencies and climate change. In (Mohr and Evans 2009) a model for forecasting worldwide coal production following three scenarios is shown. In (Fisher 2009) an analysis of past energy forecasting by EIA is done. EIA is the Energy Information Administration, the U.S. office responsible of Energy estimation. Bias in different forecast were discovered in the past estimation data, referring to a period of 22 years.

Concerning the second field, tourism, this is an appealing field for forecasting because it is expected that in the next years there will be a big increase in global demand. This mainly because there are new potential travelers, people that reached a better living conditions. In (Cho 2009) there is an analysis of future tourism demand in the Asian region. In (Smeral and Egon 2009) two scenarios of tourism demand for 2009 and 2010 are discussed, referring to the European Union context. In (Cleophas, Frank and Kliewer 2009), an interesting study about airline demand forecasting (correlated to tourism) is shown. This is a simulation tool used for determine the key performance indicator of the quality of airline demand forecasting. It generates artificial demand, so introducing a critical parameter.

The third field is the problem to forecast not regular demand, the so named intermittent demand. For such a kind of demand, time demand is lumped on few periods with peaks, while in the other periods, is zero. In this situation classical forecasting methods fails in the sense that forecast will be always zero. So other methods must be used such as (Croston 1972). Recently (Syntetos, Nikolopoulos, Boylan, Fildes and Goodwin 2009) proposed a model for including management

judgments into intermittent demand forecast. In this manner it should be possible to extrapolate useful knowledge from operators to take decisions. In (Chua, Xue-Ming, Wee Keong and Tian 2009) a new approach is proposed that is based on a variation of the Croston method. Authors shows an improvement of the forecasting ranging error from 10.22% to 27.42%.

The forth field, integration forecasting in supply chain management, is very appealing. Basic idea is that it is possible to improve supply chain management performance by integrating forecast demand with other supply chain decisions, such as inventory management.

In (Reiner, Natter, Drechsler and Wenzel 2009) the problem of shorten product life cycle is discussed. The usual assumption of stationary in this case does not apply. The author propose a system dynamics model for the integrated analysis of alternative pricing strategies and its effect on the service level. In (Bruzzone, Bocca and Poggi 2009), supermarket forecast models are developed by integrating simulation, time series analysis, and data fusion. The goal of forecasting in this case is personnel organization. The model uses time series data of sales, working hours, customer and material flows provided directly from the information system of the supermarket. Forecasts are subjected to data fusion and combined with workloads obtained from simulation. In (De Sensi, Longo and Mirabelli 2008) an analysis inventory policies under demand patterns and lead times constraints in a real supply chain operating in the beverage industry is made. Supply Chain research effort for conceptualizing, modeling, validating and simulatin it is shown.

The purpose of this paper is to find, a better estimate than for he case company, in order to improve company supply chain performances.

The interest is on the approach to use in searching the best methods more that the method itself. This because there are different product lines and a method that is good for one item maybe it is not good for other items.

In the following the case company is described with its characteristics of business, the products it covers and its future goals.

Afterwards the company supply procedure are shown, from order to delivery to the customer.

Finally the methods designed to forecast the estimated sales of a given product for 2 years, with the aim of making a comparison with the model applied by the company and verify if there are any improvements.

2. CASE STUDY COMPANY

Case considered is forecast demand for a manufacturing company in the field of office furniture. In particular company makes high quality products with a modern design and high quality materials. Main headquarters is located in New York, United States.

Other factories are in Gran Rapid, East Grenville and Toronto. In Europe there are two production site, both in Italy. Company is quality certified.

2.1. Company facilities

On the marketing point of view there are 8 business unit, that are located in:

- France;
- England (also for far east markets);
- Belgium;
- Germany;
- Switzerland;
- Unites States (main headquarter);
- Italy.

The production units are in Italy. The organization is market oriented with facilities near the business units, for maximizing client service. On the production point of view the two production units are aggregated and centralized. They are focused on three product lines.

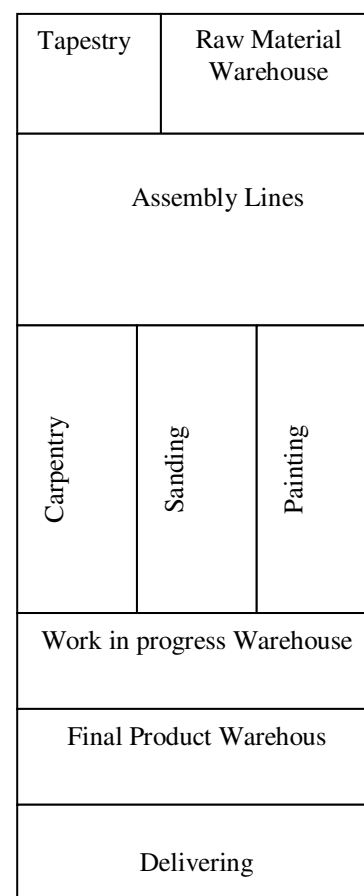


Figure 1: A layout of a company factory

The layout of production site are half product oriented and half process oriented.

As shown in figure 1, for instance, in the assembly lines there are 3 lines, each one for the assembly of one line of products. In the following, carpentry, sanding and painting department are common to all the lines.

2.2. Inventory management, Transportation and information

Regarding inventory management the policy of the company is to reduce at minimum inventories in warehouse, pushing production from make to stock towards make to order policies.

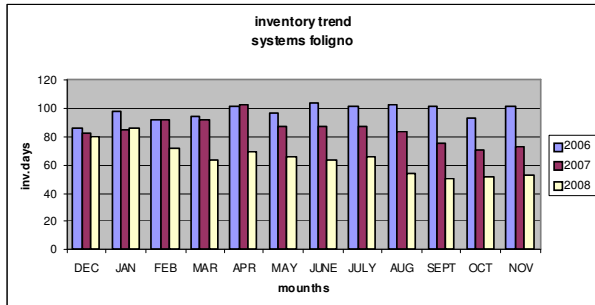


Figure 2: Inventory trends

In Figure 2 the inventory trends in the last 3 years are shown. This trend is negative, meaning this a continuous reduction of inventory.

Concerning transportation, this service is outsourced. There are three way the service is accomplished:

- Direct Shipping to the client;
- Client pick up in the business unit;
- Client pick up in the manufacturing warehouse;

Regarding the information system, the IT used allows advanced service such as integrated production planning and inventory control.

3. DEMAND FORECASTING

3.1. Demand forecasting model used by the company

Company used a moving average model on a monthly scale, with the following weight: 1 0.8 0.64 0.512 0.4096, as shown in figure 3:

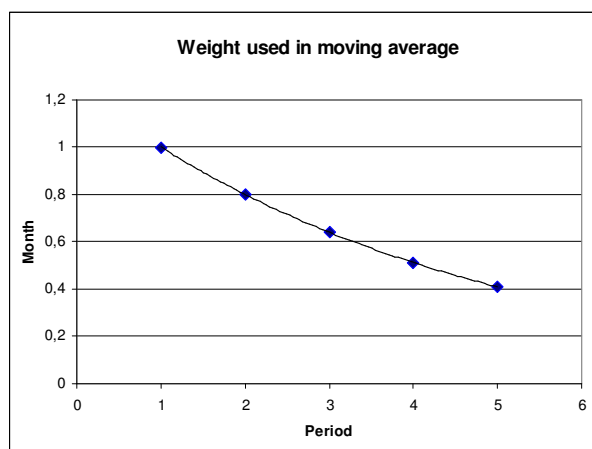


Figure 3: Weight used in the moving average

In figure 4 sales for one item for 2007 are shown. Also forecast with moving average are shown.

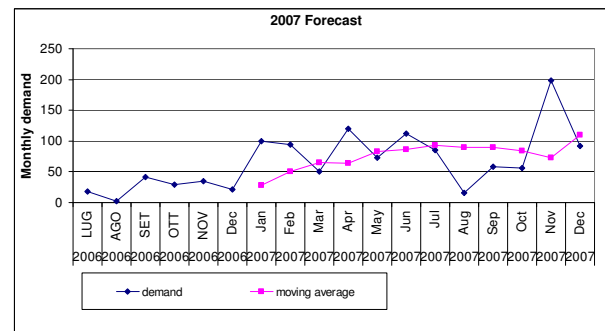


Figure 4: 2007 estimate of moving average

In table 1 results from 2 years estimation of the moving average are considered. In particular error and tracking signal are computed. This will be the base by which benchmark any other estimation method.

The estimation are based considering only one item. A more complete analysis should consider an optimization model for all the items.

First method, the static one, make a static estimation of trend and seasonality factor. This are calculated by using 2005 and 2006 data and estimation are compare with 2007 and 2008 data.

Afterwards other models are considered, exponential smoothing, different order of moving averages and the Winter and the Holt models

As shown in table 2, the Holt model gives better results, both for the Average error and for the Bias.

Second best is the exponential smoothing, that is well performing considering that is a one parameter estimator.

Afterwards moving average of order 5 give mixed results. Respect to present method is better for MAD but worst for BIAS.

It is noteworthy the underperforming of Winter Model. Probably the seasonality factor is not correctly followed by the estimator.

Also the fact that a simple moving average of order 5 performs better than the weighted moving average that was the former solution should be analyzed. There are usually updating problems when a more complicated solution is implemented. So weight that were good in a context maybe are wrong in other situations. This is one of the reasons that drives often managers towards simpler but more robust situations.

Results show a noteworthy improvement respect to the starting conditions. This it was not difficult to reach, since probably the starting point was not an optimized situation. But in any case it shows that improvements are possible in several situations. Afterwards it should be noted that analysis should be extended to more item in order to see improvement.

Even if more elaborated methods could be adopted and considered it seems not to be the case, since company behavior respect to demand forecasting is quite conservative. They prefer to follow a simple solution that a better one.

Table 1 – Estimation Results

Year	Month	Error	TS
2007	Jan	-71,5	-1
2007	Feb	-43,5	-2
2007	Mar	14,6	-2,3
2007	Apr	-56,3	-3,4
2007	May	3	-3,7
2007	Jun	-25,7	-4,7
2007	Jul	7,6	-5,7
2007	Sep	32,2	-4,5
2007	Oct	28,5	-3,6
2007	Nov	-125,4	-6,0
2007	Dec	17,9	-5,8
2008	Jan	-10,7	-6,5
2008	Feb	28,4	-5,7
2008	Mar	-4,0	-6,2
2008	Apr	-32,0	-7,2
2008	May	-4,8	-7,8
2008	Jun	49,7	-5,9
2008	Jul	-108,0	-8,2
2008	Sep	-25,8	-9,1
2008	Oct	-24,6	-20
2008	Nov	7,0	-10,0
2008	Dec	78,4	-7,3

Table 2: results of the estimation methods

	BIAS	MAD	TS
Present	-257	36,7	-7,0
Static	1317	111,0	11,9
Moving Average 5	-317,6	35,5	-8,9
Exponential Smoothing	-224,8	34,0	-6,6
Holt Model	-113,0	32,2	-3,5
Winter Model	648,0	70,3	9,1

4. CONCLUSIONS

The objective of the paper was achieved. It was possible to find a better solution compared to that of the company for the demand forecast of the selected article.

It was possible to identify precisely the approach that must be followed by any company, especially the considered one, that wants to make an estimate of future sales to ensure greater stability for production decisions.

The same analysis, therefore, should be replayed for any items to find for each one the best model.

Certainly the analysis of demand for the sales forecast is an useful tool to use, but can not be considered free from error or with absolute certainty.

It should be noted, hence, that the forecast of demand, of course, is not an exact science. There are many factors that may affect the sales of a company, such as the trend in the past, marketing items that provide period of tenders or market developments.

Especially in the period of global economic crisis, it is very difficult to predict with certainty what will be the development of demand, especially for a multinational company such the one considered, which operates throughout the world.

In conclusion it could be said that a company wishing to get good performance, able to minimize operating costs and increase earnings, must be careful about their supply chain. It should linking each one of supply chain components to be able to make better demand forecasts and possible integrate it with production planning.

REFERENCES

- Bruzzone, A.G., Bocca, E., Poggi, S., 2009. Renovating Intelligent Operations in Supermarket Chains. *Proceedings of the Third Asia International Conference on Modelling & Simulation*, 425. May 25-26 Bandung, May 29 Bali.
- Cho, V., 2009. A study on the temporal dynamics of tourism demand in the Asia Pacific Region, *International Journal of Tourism Research*, 11:465-485
- Chua, W.K.W., Xue-Ming Yuan, Wee Keong Ng, Tian Xiang Cai, 2008. Short term forecasting for lumpy and non-lumpy intermittent demands, *Proceedings 6th IEEE International Conference on Industrial Informatics*, 1347. July 13-16, Daejeon, Korea.
- Cleophas, C., Frank, M., Kliewer, N., 2009. Simulation-based key performance indicators for evaluating the quality of airline demand forecasting. *Journal of Revenue & Pricing Management* (1476-6930) 8:330-342.
- Croston, J.D., 1972. Forecasting and stock control for intermittent demands. *Operational Research Quarterly* 23:289-303.
- Day, A.R., 2009. Forecasting future cooling demand in London. *Energy and Building* 41:942-948
- De Sensi G., Longo F., Mirabelli G., 2008. Inventory policies analysis under demand patterns and lead times constraints in a real supply chain. *International Journal of Production Research*, 46: 6997-7016.
- Fischer, C., 2009. Understanding errors in EIA projections of energy demand. *Resource and Energy Economics* (0928-7655) 31:198-209.
- Hahn, H., 2009. Electric load forecasting methods: Tools for decision making. *European Journal of Operational Research* 199:902-907.
- Mohr, S.H., Evans, G.M., 2009. Forecasting coal production until 2100. *Fuel*, 88:2059-2067.
- Reiner, G., Natter, M., Drechsler, W., 2009. Life cycle profit-reducing supply risks by integrated demand management. *Technology Analysis & Strategic Management* (0953-7325) 21:653-664.

- Smeral, E., 2009. The Impact of the Financial and Economic Crisis on European Tourism. *Journal of Travel Research* (0047-2875) 48:3-13.
- Syntetos, A., Nikolopoulos, K., Boylan, J. E., Fildes, R., Goodwin P., 2009. The effects of integrating management judgement into intermittent demand forecasts. *International Journal of Production Economics* 118:72

supporting decision tool in the assembly line balancing problem") through genetic algorithms; discrete event simulation; Interactions between Demand Forecasting and Inventory control policies; Lateral shipments policies in multi-echelon networks.

AUTHORS BIOGRAPHY

Stefano SAETTA is associate professor at the Industrial Engineering Department of the University of Perugia since 22/12/2004. He obtained the Electronic Degree in 1991 summa cum laude at the University of Perugia where he became assistant professor in 1995. He teaches courses: Industrial Plants for mechanical engineers, Industrial Plants, Management of Logistic systems, Logistics. He was visiting professor at Rutgers University, NJ, USA in 2005 and at the Arizona University, AZ, USA in 2007 and 2008. His research field covers: modelling and simulation of logistic and productive processes, methods for the management of life cycle assessment, discrete event simulation, supporting decision methods, lean production. He was involved in several national and international research projects such as: ISACOAT (2001-2004)- Network on Integrated Scenario Analysis of Metal Coating, as co-responsible, supported by The European Union within V research Framework; GIPIETRE (2005-2006), as responsible, supported by ICE, an Italian agency for international trading and in cooperation with companies from Brazil and China; VOC-LESS (2006-2009) responsible for the Perugia Unit of the project VOC less pulping, supported by European Union within the Life framework. He is responsible of many research projects of his department, funded by Italian ministry of university or by other entities. He is active in ERASMUS exchange programs with the University of Marseille and Salford University. His Internationally Affiliated Faculty of the Industrial and Systems Engineering Department of the Rutgers University. His the director of MISS- McLeod Institute of Simulation Science - Perugia Center and International Co-Vice Director. He is author of more than 70 publications.

Lorenzo TIACCI is assistant professor at the Department of Industrial Engineering of the University of Perugia. He obtained the degree summa cum laude in Mechanical Engineering in 1998 and the Phd in Industrial Engineering at the University of Perugia in 2004. He teaches Inventory Management and Production Planning & Scheduling at the Faculty of Engineering – University of Perugia. He worked as consultant in many research projects on production planning and control, logistics and operations management, in cooperation with research centers and companies. His research fields of interest are: production system design, planning and control; assembly line balancing (argument of my Phd final dissertation: "A simulation based methodology as

RISK ANALYSIS FOR INDUSTRIAL PLANTS PROJECTS: AN INNOVATIVE APPROACH BASED ON SIMULATION TECHNIQUES WITH EXPERIMENTAL ERROR CONTROL

^(a)Roberto Mosca, ^(a)Agostino G.Bruzzone, ^(a)Lucia Cassettari, ^(a)Marco Mosca

^(a)DIPTeM Department of Industrial, Production & Thermoenergetics Engineering and Mathematical Models

^(a)roberto.mosca@diptem.unige.it, ^(a)agostino @itim.unige.it, ^(a)cassettari@diptem.unige.it,
^(a)marco.mosca@liophant.org

ABSTRACT

This paper illustrates an innovative method for economic risk analysis tailored for being applied to industrial plant project. The proposed method allows to estimate the total "Level of Coverage" of the plant projects, in other words the probability that identified risks don't reduce the overall economic result below zero even considering the value of total contingency funds allocated, supposing that these one are adequate in compensating extra costs due to risk sources and events.

The procedure involve several steps, the first one is focusing on the determination of the probability distribution associated with the enterprise's various costs for each individual risky event considering its specific economic impact.

As following step a Monte Carlo simulation campaign is conducted to experimentally measure the final value of the Level of Coverage.

In the paper it is proposed the application for demonstrating the effectiveness of the approach illustrated herein in reference to a case study related to risk analysis in complex industrial plant project for railway sector (Bruzzone and Orsoni 2000)

Keywords: Monte Carlo Simulation, MSPE

1. INTRODUCTION

The risk management process for industrial project starts from the quotation phase, when the main potential criticality is to be identified/quantified and the budget including profit and losses of the project are estimated. Therefore, once the project quotation phase is started, a detailed risk management process needs to be developed (Bruzzone et al. 1999)

The process under consideration envisages an activity for the identification of risks and their sources, the assessment of these risks including all possible mitigation actions in terms of effectiveness and costs and the determination of the proper contingency level to be allocated for facing identified risks.

Of course, since mitigation actions affect the common sources of different types of risks, it is important to plan these actions with a cross-the-board approach to create synergies at a task, program,

business unit and company aggregate level (Pierce 1998)

The risk management process (Schenone et al. 1999) must also be constantly updated as significant events occur and risk estimations are evolving.

The three main phases of a correct risk management process are listed below:

- Risk identification: using a checklist, all of the possible risks, which may have an impact on the project, are identified.
- Risk assessment and exposure reduction: risk reduction by means of specific mitigation activities involves simple criterions for assessment of the effectiveness of the different actions based on the net benefit analysis to determine economic cost-effectiveness.
- Risk monitoring and review: to monitor the trend in contingency allocations over time

The depth and the relevant formalization of the analysis depend on project relevance in terms of complexity and strategic importance and have to be assessed by determining the project class of risk (A, B, C).

An in-depth analysis of risk issues must be carried out by taking into account the general risks of the context and the specific ones associated with the relationship between the services required and the company's capacity.

The paper provides a procedural description of these three phases devoted to complete risk analysis.

2. RISK IDENTIFICATION PHASE

The first phase aimed at identifying all possible risks which could affect the project under consideration (Navarrete 1995) In order to determine the above mentioned risks, a checklist was used as a guide. This made it possible to identify different types of risks (Nepi 2007). In particular, it should be noted that there are two types of risks: those which are purely operational, that is strictly linked to activities and those of a legal/financial nature relating to contract clauses such as, for instance, those regarding penalties. The checklists under consideration were filled out with interviews for the personnel directly

involved in the project management (Arcibald 1994, Duncan 1996); the related data was then summarized and re-organised. Alongside the identification of the risks, the WP's (Work Packages) affected by the risks were also identified to facilitate both the subsequent quantification phase, which obviously depends on the WP's affected by the risk, and the contingency allocation phase. After having completed the first phase and obtained a list of the risks deemed to be most significant, these were quantified as illustrated in the figure 1.

3. RISK ASSESSMENT AND EXPOSURE REDUCTION PHASE

Once having identified the most significant risks, an assessment of these was then performed. Firstly, two types of analysis were performed: a qualitative and quantitative analysis respectively (Bruzzone 1998). The qualitative analysis consisted in classifying the single project risks to determine a priority in their management. The quantitative analysis involves a quantification of the economic impact to be performed if critical risky event happen (Goodpasture 2003). Having identified the risks and performed the qualitative and quantitative analysis of the risks, the possible actions have to be identified to mitigate their impact. This activity was supported by an analysis of the causes generating the single risks for the purpose of reducing/eliminating the risk's impact. In fact the implementation of a mitigation action allows to reduce a potential cost (risk regulated by stochastic behavior) by incurring in a lower certain cost (the cost of the very action). The cost of these actions was considered a "certain cost" and, as such, it must be posted in the profit and loss account. In order to ensure the proper balance between the reduction in risk and the cost-effectiveness, it was necessary to determine the net benefit of the mitigation actions. This benefit was determined as the difference between the expected value of the risk before and after the mitigation action without the costs for these actions. Clearly only the mitigation actions generating a positive net benefit have to be implemented unless there is anyhow a general benefit for the project and/or company (Willoughby 1995)

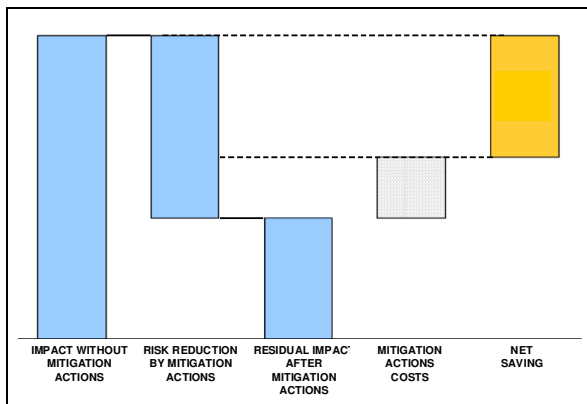


Figure 1 - Calculation of the net benefit of the mitigation action

Therefore, with regard to the study, as stated above, following the identification and assessment of the mitigation actions, those risks deemed significant were analyzed once again in terms of impact and probability and any contingencies to be allocated were defined considering the effect of the mitigation actions in the light of the residual aleatory component of risk.

Downstream of the quantitative analysis of the risks and determination of the mitigation actions to be adopted, the amount of the contingencies to be allocated for the residual risks was defined.

The allocated amount must duly "cover" the project if the feared risk should actually occur.

In this study it was decided to allocate contingencies in the amount of the value of the estimated impact, trying to minimise any possible extra costs associated with the risk. In particular, the percentage of contingencies to be allocated was selected based on the probability that the risk occurs, according to the criteria in Table 1:

Table 1 – Correspondence between the occurrence probability and the amount of the contingency to be allocated

CONTINGENCY	
Prob ≥ 50%	Contingency = Entire value of impact
Prob ≤ 20%	Contingency = 30% of entire value of impact
20% < Prob < 50%	Contingency = 50% of entire value of impact

In the following phases of economic re-forecasting related to the project in progress or update of the risk analysis, the described process has to be updated. All the risks which do not occur, must be analyzed again in order to re-determine the relevant associated contingency, if any.

4. THE MONTE CARLO SIMULATION APPROACH

Classical risk quantification methods are based on the assumption that the expected risk value, calculated as the product between probability and impact, is capable of determining the essential aspects of the project overall risk profile.

However, in some cases, it may be useful to perform further analyses taking into account those aspects linked to the fact that risk may occur for multiple values of the economic impact. Based on the above, the analysis under consideration herein makes possible to take into account aspects linked to the project overall risk profile in order to define the project overall level of coverage for a given amount of contingency.

The adopted procedure (Bruzzone et al. 2005) consists in identifying, for each risk, the probability distribution of the different values of the impact of the risky event. If the latter can be determined with economic impacts marked by differing levels of

severity, the use of probability distribution helps to properly represent the occurrence probability for each. (Bruzzone et al. 2004, Bruzzone et al. 2007)

The various numerical simulation software currently available, make possible to rapidly and easily define the probability distributions simply by selecting and customizing set models. These characteristics led us to choose one of these applications to develop the proposed analysis, i.e. Crystal Ball.

Based on the previous data, it was possible to determine the total value of contingency to be allocated to cover the risks considered. Therefore, by calculating the “confidence level” of the total contingency allocated, it was possible to assess the consistency of the coverage, compared to the probability of having to incur the extra costs of the risks expected when these occur.

“Confidence level” is understood as the probability that the risks occur during the project with an overall impact lower than the contingencies allocated and hence that these are adequate in preventing a negative impact on the margin.

In order to calculate the Contingency Level, two operations are needed:

1. characterization of the risks by assigning to each a probability distribution describing the possible trend in the impact values, which can be assumed from the risk itself;
2. performance of a Monte Carlo simulation, using a statistical analysis software (in this case, Crystal Ball) in order to determine the project overall probability distribution starting from those assigned to the single risks.

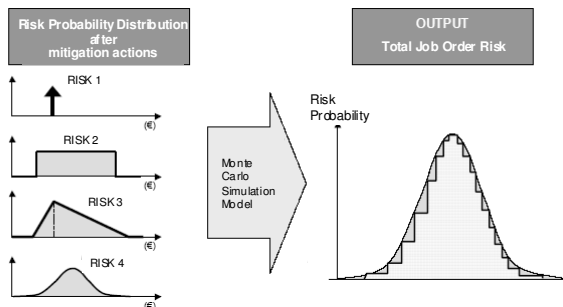


Figure 2- Example of calculation of project overall probability distribution

The Monte Carlo analysis (Mosca et al. 2008) makes possible to determine of the probability distribution of the costs linked to the occurrence of project risks and the degree of coverage corresponding to each value of the overall contingency allocated, through the representation of the cumulative curve of the probabilities.

The confidence curve obtained by integrating the probability distribution shows on the x-axis the possible contingency values and the “Confidence Level” on the y-axis. The Confidence Level is “the

probability of succeeding in covering in full the costs resulting from the occurrence of the risks using only the overall contingency allocated.”

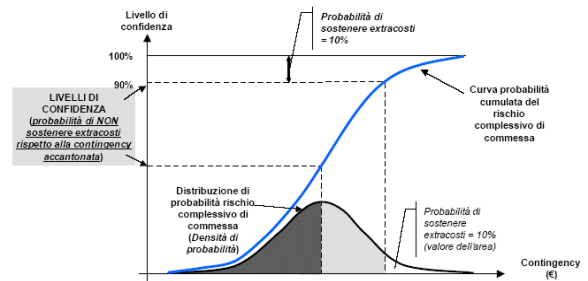


Figure 3- Example of confidence level and cumulative probability curve

5. APPLICATION TO AN INDUSTRIAL CASE

We now illustrate the results obtained from the application of the described method to an industrial project related to an automation project in transportation (Bruzzone et al. 2000; Bruzzone and Mosca 2003)

Once the risks were selected as mentioned above, each was associated with an appropriate probability distribution. In particular, beta distribution was chosen for the 17 risks with an economic impact not equal to zero. As for the risks with an economic impact not equal to zero, since the value is the result of a mitigation action involving the risk's elimination, it was deemed appropriate not to assign any probability distribution. Through the selected beta probability distribution, it was possible to assign a greater occurrence probability to the impact value considered most likely and, at the same time, assign anyhow a greater occurrence probability to those values lower than the estimated impact compared to those with higher values (the company deemed such a decision proper, on the basis of an accurate a priori estimation of the impact itself). The need to resort to this type of distribution was determined by the lack of historical reference data allowing a more accurate distribution.

Once the required parameters were set, 20000 Monte Carlo simulation runs were carried out and these gave the total contingency trend and the Confidence Level associated with the project.

Some relevant graphs are provided below (Fig. 4 and 5):

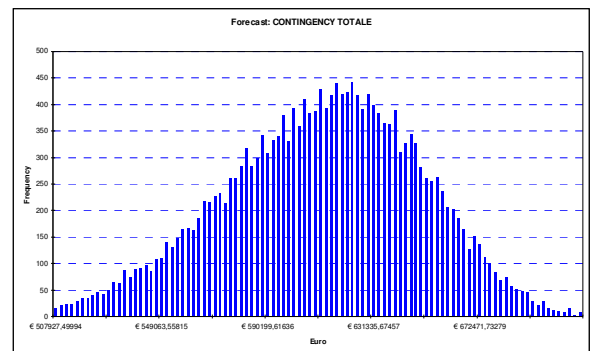


Figure 4 - Total Contingencies Trend

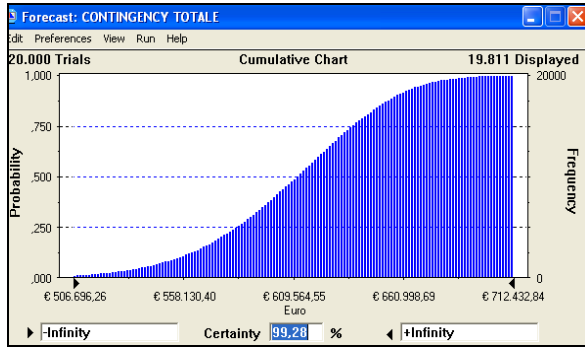


Figure 5- Confidence Level

As shown in Fig. 4, the contingencies tend to spread according to a normal trend whose most probable value stands at about the value corresponding to the estimated total contingency obtained in the analysis expounded above. The information obtained from the graph in Fig. 5 is definitely much more interesting, as it tells what the probability is of having extra costs once a certain contingency level is set. In the case under consideration, for instance, a total estimated contingency value (€ 653,665.65) gives a probability of coverage of about 80%, which hence corresponds to a 20% risk of having extra costs. It should be pointed out that said graph does not provide information on the value of the extra costs, which will be incurred, but said value will clearly vary depending on the contingencies accrued. In the considered case, for instance, allocations in the amount of € 500,000 would give a risk of extra costs of almost 100% as would be the case if no allocation were made, but clearly the extra costs, which would be incurred in both cases, would be much different.

6. VALIDATION TEST OF THE SIMULATION RESULTS

Finally, it is necessary to validate the conclusions by using the analysis method of the Mean Square Pure Error evolution in the replicated runs [3]. This method makes possible to understand when the model output tends to stabilize due to the system's growing lack of sensitivity to the new casual draws. The MSPE curve must have the traditional knee curve with an asymptotic trend along the x-axis and decreasing punctual disturbances as the simulated time increases. During the validation phase, 5 series of simulations, each of 20000 replicated runs, were performed.

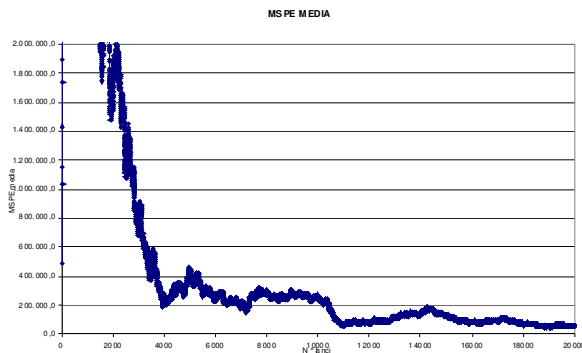


Figure 6 - MSPE_{MED} evolution curve

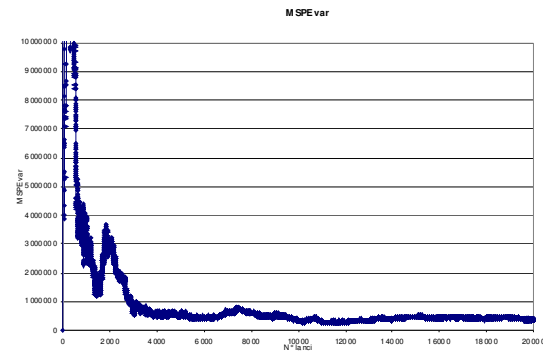


Figure 7 - MSPE_{STDEV} evolution curve

Based on the simulation results, a calculation of the punctual values of the MSPE_{MED} and MSPE_{STDEV} was performed: the curves shown in Fig. 6 and 7 arose from that.

The analysis of the two graphs shows that after about 20000 runs the experimental error is stable and reaches a level compatible with an error bandwidth such as the value of the mean response is not disturbed.

Applying the formula below in order to calculate the error bandwidth, i.e. the interval within which 99.6% of possible simulation outputs are found, at 20000 runs:

$$y = \bar{y} \pm (3\sqrt{Mspe_{MED}} + 3\sqrt{VAR + Mspe_{STDEV}}) \quad (1)$$

the following range of variability of the punctual response was obtained:

$$€ 490,394.83 < y < € 729,909.53$$

with:

Table 2: MSPE

\bar{y}	610.152,18 €
$MSPE_{MED}$	58.056,0 € ²
$MSPE_{STDEV}$	378.455,6 € ²
VAR	1.573.978.593 € ²

Finally, a check of the correctness of the identified range, was carried out considering the 10000 simulation runs carried out to build the curves and calculating the percentage of responses falling within the range. About 370 values fell outside the set interval, which is in line with expectations, as it corresponds to about 0.4% of the runs.

7. CONCLUSIONS

The transition from a deterministic regimen to a stochastic one, in analyzing industrial risks, represents a major milestone in the effort to find a high degree of correspondence between the model results and the operational reality.

The Monte Carlo method is an effective methodology for dealing with the stochastic elements affecting in real systems. However, decision-makers (Kimmons 1990, Kliem et al. 1997, Kerzner 1998,) need to be cautious during the model experimentation phase in order to avoid the risk of obtaining an incorrect number of runs due to improper choices in probability distributions for describing the various risks. (Harrison and Lock 2004). Based on the proposed method it is possible to know with an adequate degree of accuracy the number of experimental replications needed to obtain an output within a predefined degree of reliability (Bruzzone and Revetria 2002).

As for the proposed application, the authors would point out that the investigated method made possible to determine the proper range for contingency funds to be allocated and estimate the probability of residual risks within predefined reliability levels.

REFERENCES

- Arcibald R.D. (1994) "Project management. La gestione di progetti e programmi complessi", Franco Angeli, Milano
- Bruzzone A., Giribone P., Mosca R. & Fassone B. (1999) "Project Management Tools: Simulation On Web Environment", Proceeding of Websim99, San Francisco, January
- Bruzzone A.G. (1998) "Gestione dei Progetti d'Impianto", DIPTeM Technical Report, University of Genoa, Genoa, Italy
- Bruzzone A.G., Frydman C., Giambiasi N., Mosca R. (2004) "International Mediterranean Modelling Multiconfernece", DIPTeM Press, ISBN 88-900732-4-1 Vol I e II (884 pp)
- Bruzzone A.G., Giribone P., Revetria R. (2000) "Simulation as Educational Support for Production and Logistics in Industrial Engineering", Proceedings of WinterSim, Orlando, December
- Bruzzone A.G., Longo F., Merkurjev Y., Piera M.A. (2007). International Mediterranean And Latin America Modelling Multiconference. ISBN: 88-900732-6-8. GENOVA: DIPTeM Press (ITALY)
- Bruzzone A.G., Mosca R. (2003) "Modelling & Applied Simulation", DIPTeM Press, Genoa, ISBN 88-900732-3-3 (197pp)
- Bruzzone A.G., Orsoni A. (2000) "Advanced Modelling for Multilevel Analysis in Complex Processes Involving Ship Construction" JORBEL (Journal of the Belgian Operations Research Society), Vol. 40, no. 3-4, pp. 219-234 ISSN:0770-0512
- Bruzzone A.G., Reverberi A.P., Maga L., Barbucci (2005) "Montecarlo Simulation of a Ballistic Selective Etching Process in 2+1 Dimensions", Physica A 354, 323-332 - ISSN: 0378-4371
- Bruzzone, A. G. Revetria, R. (2002) "Reliability Analysis by Using Simulation for Complex Automated Plants", SIMULATION SERIES 2002 VOL 34; PART 4, page(s) 115-119
- Society for Computer Simulation, ISSN-0735-9276
- Duncan W. (1996) "Project Management Body of Knowledge", Project Management Institute, Charlotte, NC, USA
- Goodpasture J.C. (2003) "Quantitative methods in project management?", J. Ross Publishing
- Harrison F.L., Dennis Lock, (2004) "Advanced project management: a structured approach"?, Gower Publishing, Ltd.,
- Kerzner H. (1998) "Project Management", Van Nostrand Reinhold, New York
- Kimmons R.L. (1990) "Project Management Basics", Marcel Dekker Inc.
- Kliem R.L. Robertson K.L. Lundin I.S. (1997) "Project Management Methodology", Marcel Dekker Inc.
- Mosca R., Giribone P., Revetria R., Cassettari L., Cipollina S. (2008) "Theoretical Development and Applications of the MSPE Methodology in Discrete and Stochastic Simulation Models Evolving in Replicated Runs, Journal of Engineering", Computing and Architecture ISSN 1934-7197, Volume 2, Issue 1
- Navarrete P.F. (1995) "Planning, Estimating and Control of Chemical Construction Process", Marcel Dekker Inc.
- Nepi A. (2007) "Project Risk Management, Analisi e gestione dei rischi di progetto", FrancoAngeli, Milano
- Pierce D.R. (1998) "Project Scheduling and Management for Construction, Means Co, Kingston MA
- Schenone M., Frigato A., Mosca M. (1999) "Risk analysis applied to project management and project control" Proc. XIX IASTED International Symposium "Modelling, Identification and Control" - MIC'99, Innsbruck, A
- Willoughby (1995) "Project management for Builders & Contractors", Salal Press

AUTHORS BIOGRAPHY

Roberto Mosca is Full Professor of "Industrial Plants Management" and "Economy and Business Organization" at the DIP (Department of Industrial Production & Engineering, former Institute of Technology and Mechanical Plants), University of Genoa. He has worked in the simulation sector since 1969 using discrete and stochastic industrial simulators for off_line and on_line applications. His research work focuses on the evaluation of simulation languages and new modeling techniques and his research team is developing new AI applications for industrial plant management. Currently he is directly involved as coordinator in the construction of a new campus in Savona, focused on industrial engineering. He was Director of ITIM (Institute for Technology, Industrial Engineering and Manufacturing, Istituto di Tecnologie ed Impianti Meccanici), DIP (Department of Industrial Production, Dipartimento di Ingegneria della Produzione) and DIPEM (Department of Industrial Production, Engineering and Mathematical Modelling), Dipartimento di Ingegneria della Produzione e Modelli

Matematici) in Genoa University. Currently he is Director of DIPTeM (Department of Industrial Production, Technology, Engineering and Modelling), Dipartimento di Ingegneria della Produzione, Termotecnica e Modelli Matematici) in University of Genoa.

Agostino G. Bruzzone since 1991, he has taught "Theories and Techniques of Automatic Control" and in 1992 he has become a member of the industrial simulation work group at the ITIM University of Genoa; currently he is Full Professor in DIPTeM. He has utilized extensively simulation techniques in harbor terminals, maritime trading and sailboat racing sectors. He has been actively involved in the scientific community from several years and served as Director of the McLeod Institute of Simulation Science (MISS), Associate Vice-President and Member of the Board of the SCS (Society for Modelling & Simulation international), President of the Liophant Simulation, VicePresident of MIMOS (Movimento Italiano di Simulazione) and Italian Point of Contact for the ISAG (International Simulation Advisory Group) and Sim-Serv. e has written more than 150 scientific papers in addition to technical and professional reports in partnerships with major companies (i.e. IBM, Fiat Group, Contship, Solvay) and agencies (i.e. Italian Navy, NASA, National Center for Simulation, US Army). He teaches "Project Management" and "Industrial Logistics" at the University for students in the Mechanical Engineering (4th year), Management Engineering (4th year) and Logistics & Production Engineering (3rd Year) Degree Courses.

Lucia Cassettari earned her degree in management engineering in 2004 at the University of Genoa. Currently she works in the Department of Production Engineering, and Mathematical Modelling (DIPTeM) at the University of Genoa as a researcher, in the field of simulator-based applications for industrial plants; particular attention is focused on the application of DOE and Optimization techniques to industrial plant problems using Simulation. She is also active in research projects involving BPR (Business Process Reengineering), Activity Based Management and Cost Control. She is senior lecturer in the Industrial Plant Management and Business Strategy degree courses.

Marco Mosca earned his degree in Management Engineering in 1999 at the University of Genoa. He has more than 10 years of working experience, consultant in Accenture and manager in Bombardier Group working in Italy, Switzerland, Austria and Germany; currently he is director of a Manufacturing .company in a South Europe; he attended to international conference as speaker and he has several papers on researches developed in joint cooperation with DIPTeM University of Genoa.

SIMULATION AND ANALYSIS OF A PRODUCTION LINE WITH MULTIPLE OUTPUT FORMATS

Andrea Grassi^(a), Giuseppe Perrica^(b), Elisa Gebennini^(c), Sara Dallari^(d), Cesare Fantuzzi^(e), Bianca Rimini^(f)

Dipartimento di Scienze e Metodi dell'Ingegneria
Università degli Studi di Modena e Reggio Emilia
Via Amendola 2 – Pad. Morselli, 42122 Reggio Emilia, Italy

^(a)Andrea.Grassi@unimore.it, ^(b)Giuseppe.Perrica@unimore.it, ^(c)Elisa.Gebennini@unimore.it,
^(d)Sara.Dallari@unimore.it, ^(e)Cesare.Fantuzzi@unimore.it, ^(f)Bianca.Rimini@unimore.it

ABSTRACT

This paper deals with simulation based performance analysis of production lines characterized by the capability to produce multiple product formats at the same time. This situation is typical of several line production systems, mainly in packaging lines where the output consists in different formats of secondary packages containing a different amount of single products. The line is then configured in a first part that works single units of product, and in a second part in which the production flow is split into different branches where specific machines arrange single products in the format needed by the requested secondary package. In the paper, which is derived from an actual case study, key factors of such kind of lines are analyzed and their effects on line performance are investigated. Some crucial aspects and guidelines are then pointed out to constitute a support in the line design phase.

Keywords: simulation, manufacturing systems, systems design, productivity, throughput.

1. INTRODUCTION

In its simplest form a production line is composed by stations and buffers. Stations execute processing tasks on units, whereas buffers consists in storage areas for accumulating incomplete manufactured units between consecutive stations. This basic elements are joined together according to serial and/or parallel connections. Since the machines are close coupled each others, the performance of the whole line depends strongly on both the performance of single machines and how they are linked. As a matter of fact, the line throughput is discontinuous as a consequence of disruption of the production flow that can happen for different reasons: given that machines are unreliable failures occur, product jamming can stop the flow, but also raw materials replenishment may require machine stop.

Flow splitting equipments can also be present in a manufacturing system. They are usually adopted to divert the production flow to different ways to produce

different output solutions starting by the same basic product. A production line with a splitting equipment can be viewed in two blocks: a first part composed by serial connected machines and a second one composed by two or more branches of serial connected stations. Final product diversification is realized in this second block of the whole line. Furthermore, it is important to notice that control policies imposed to the splitting device can reduce losses in line throughput when one of the branches in the second blocks is jammed. Buffers are other components helpful to prevent the overall system against rapid disruption propagation, thus contributing to sustain the overall production flow. A key design parameter for a buffer is its size: the longer failures to cover are, the larger its capacity must be. If machines are characterized by high production rate, it is easy to understand that buffer size required could be unfeasible in practical applications.

The most popular approaches to study unreliable production lines can be classified in analytical techniques or stochastic discrete event system based methodologies. In literature, various continuous-flow and discrete-flow material models have been proposed for transfer lines under the markovian assumption for machines reliability characteristics. Numerous analytical approaches have been proposed for approximate performance evaluation (Dallery, David and Xie 1989; David, Xie and David 1990; Alvarez-Vargas, Dallery and David 1994; Fu and Xie 2002; Dallery and Gershwin 1992). Moreover, methods for analyzing systems with splitting/merging components have also been presented (Gershwin and Burman 2000; Diamantidis and Papadopoulos 2006). These models provide rapid solutions for different line configurations, but they are generally not able to describe some specific aspects of real lines behavior, such as raw material consumption. In addition, the assumption of exponential distribution for time to failure and time to restoration modeling, and the moderate detail level of these models, makes them suitable in the first design phase.

Discrete event simulation is another technique for analyzing complex production lines, being able to take

into consideration high detail level, but not able to provide an immediate understanding of the system behavior and requiring more computational time if compared with an analytical model. Several examples can be found in literature (Curcio, Longo and Mirabelli 2007; Harrell and Gladwin 2007; Gujarathi, Ogale and Gupta 2004; Patel, Ashby, and Ma 2002). Simulation approach is a highly effective tool for the design of a manufacturing system given its ability to meet design goals considering the typical complexity of actual systems. Time to failure (TTF) and time to repair (TTR) profiles can be modeled by means of whatever statistical distribution, and a consistent detail level could be included, if needed, in system description.

In this paper a soap manufacturing line has been investigated by using discrete event simulation approach. The analyzed system is a typical case representative of those several industrial production systems in which high production rate is required together the need to produce different formats at the same time. Furthermore, raw material replenishment has a sensitive impact on line operation. Final product diversification is realized by means of two branches, in the second block of the line, each of them presenting a bounding machine dedicated to arrange single items in a specific package format. The two branches can have different production rates, hence, if one is in blocking state, line throughput is reduced differently.

The main contribution of this work is to provide general considerations and design guidelines for such complex systems. The findings concern with the influence of the most important design factors and line configuration on line throughput. Investigated scenarios analyze line productivity for several buffer size, different MTTF and speeds of downstream stations. Moreover, automatic and manual solutions for film splice operations in bundling machines have been compared.

This paper is organized as follows: Section 2 discusses the addressed problem, model assumptions and description are provided in Section 3, while Section 4 reports experimental campaigns configuration and provides results analysis. Finally, Section 5 points out concluding remarks.

2. PROBLEM DESCRIPTION

The system considered in this paper is a soap packaging line able to arrange soaps into two different formats at the same time. For this primary reason, two bounding machines are put in the final part of the line. Each of them is then configured accordingly with the format needed by the production plan.

The initial part of the line is constituted by a series of two machines, the stamper and the cartoner, to form soaps and to put them into a carton package.

The stamper works by forming ten soaps (one rank) in a cycle time and has a nominal capacity of 600 soaps per minute. To improve synchronization, and thus to reduce jamming, the stamper and the cartoner are connected in a rigid manner (i.e. by means of a box-belt).

The cartoner brings soaps from the box-belt and put them individually into carton packages. This machine has a nominal capacity equal to the one of the stamper.

In the immediate downstream a divider is placed, so as to split items flow to the two bounding machines. The control policy of the divider depends from the capacity of the bounding machines, as explained in the sequel.

The bounding machines produce secondary packages by arranging soaps in a specified format and wrapping them with a slim plastic film. The nominal speed of the machine is also related to the number of soaps in the secondary package: the higher the number of soaps in the package, the less the working time needed to form the package tends to be. Another important aspect of such machines is the time needed to change the film drum and to splice the new film with the ending part of the currently loaded one. When manually executed, such a task is consistently time consuming (some minutes) and imposes a stop of the machine. Most recent bounding machines can be equipped with an automatic film splice unit that makes it possible to execute the film change task in masked time, thus avoiding the need to stop the machine.

In the examined case, the first bounding machine is set up to produce a format containing 6 soaps, and can work at a speed of 430 soaps per minute. Since it is a mature model, it is not equipped with an automatic film splice unit, thus film change must be executed manually. The second bounding machine is set up to produce smallest packages. In this study, secondary packages formed by two soaps are addressed since they imply the slowest speed for the machine. This second bounding machine is a newer model and is equipped with an automatic splice unit for automatic film change. Two different nominal capacity are addressed in the study: 200 and 400 soaps per minute.

Buffers can also be included in the line in such a way as to sustain production flow when some machines are down for jamming reasons or as a consequence of manual splice operations on bounding machines. Since the addition of buffers involves an increase in costs, and also considering that the high speed of such kind of lines suggests the adoption of large sized buffers, no more than one buffer can be conveniently added in the line, this for economic reasons.

Hence, issues concerning location and sizing of the buffer constitute an important aspect. For what concerns the buffer location problem, it has to be pointed up that some constraints must be taken into consideration. First of all, the buffer cannot be located between the stamper and the cartoner since, as said before, they must be perfectly synchronized to reduce jamming in the cartoner itself. Hence, the buffer can be located immediately before the divider, or immediately after it in one of the two branches.

Since the divider has a nominal maximum speed aligned with the one of the stamper, if the buffer were placed immediately before the divider a potential overcapacity of the bounding machines could not be exploited. In other words, the divider would act as a bottleneck each time a capacity of the bounding

machines greater than the capacity of the stamper is installed. This is the reason why the buffer will be positioned after the divider, in the examined case.

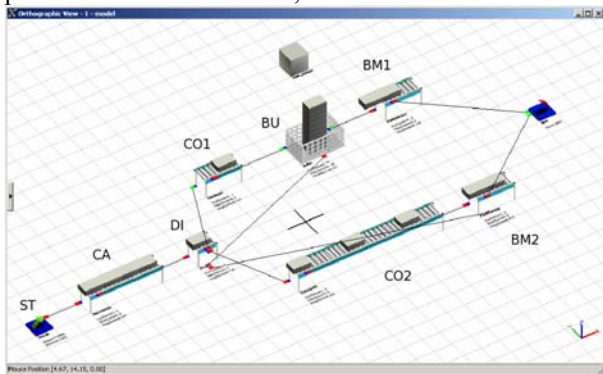


Figure 1: Simulation model.

Different control policies can be adopted for the divider dependently by the configuration of the two bounding machines.

When a second bounding machine with a nominal speed of 200 soaps per minute is considered, the only policy suitable for the divider is to generate a flow towards the first bounding machine that is twice the flow towards the second bounding machine. Since the divider, for mechanical reasons, needs to send at least ten soaps to an output before to switch to the other output, a policy that sends twenty soaps to the first bounding machine each ten soaps sent to the second bounding machine is adopted.

On the contrary, if a second bounding machine with a nominal capacity of 400 soaps per minute is adopted, the divider can be configured to switch the biggest part of the flow towards the second bounding machine each time the buffer is almost full. In this way, a nominal capacity of 800 soaps per minute can be reached in the downstream.

When jamming occurs in the upstream of the line (i.e. stamper, cartoner, or divider), the buffer can sustain the production of the first bounding machine, that is, the downstream can work at a speed of 400 soaps per minute until the buffer goes empty. When jamming occurs in the first bounding machine, the divider maintains its default policy until the buffer accepts soaps. When one of the two branches cannot accept soaps cause to jamming or to buffer full conditions, all the flow is diverted to the operative branch.

The stamper is able to slow down its speed depending on the state of the downstream, and it can also be put in full recycling mode each time the downstream is completely blocked.

A final important aspect involving bounding machines behavior is the presence of an automatic splice unit able to automatically change the film, thus avoiding the need to stop the machine. The first bounding machine is not equipped with an automatic splice unit, hence the film change task must be executed manually causing the need to stop the machine for some minutes. Each time a manual film change happens, the buffer faces a noticeable amount of workload and can go full. For that reason, the effect of the adoption of the

automatic splice also in the first bounding machine was investigated.

3. SIMULATION MODEL

The simulation model, shown in Figure 1, was built using the Flexsim™ environment. Table 1 reports the description of the objects characterizing the model.

Some simplifications were made to speed up model run time, and some tricks were adopted to correctly describe the behavior of the machines.

First of all, each flow item moving in the model represents a set of ten soaps (the equivalent of one rank produced by the stamper), and its length is representative of the length occupied on conveyors by the set of soaps. This allows to consistently speed up simulation runs without losing in analysis capability.

Table 1. Objects in the model

ST	Stamper
CA	Cartoner
DI	Divider
CO1	First conveyor
CO2	Second conveyor
BU	Buffer
BM1	First bounding machine
BM2	Second bounding machine

The stamper ST was modeled as a flowitem generator and was set up to produce at the nominal capacity of 600 soaps per minute.

The cartoner CA was modeled as a conveyor to correctly represent the internal accumulation of 40 soaps of the real machine. The conveyor speed was set up in the model to reproduce a nominal capacity of 600 soaps per minute.

Also the two bounding machines BM1 and BM2 were modeled using a conveyor object. This choice was made to overcome a typical problem that affects time to failure consumption computation in processing objects that can vary their production speed. Since failures (jamming) are operation dependent, time to failure consumption must be computed only when the machine is in the operating state. In a simulator, the typical processor object used to model processes on flow items has an operating cycle time that depends from the speed. The high the speed, the less the operating time spent by the processor on the flow item. Hence, if the flow entering the processor object is constant (i.e. it is imposed by the upstream), as the speed of the processor increases, its reliability increases since the time spent in the operating state per flow item is reduced.

For that reason, the two bounding machines were modeled using conveyor objects, and the consumption of the time to failure was related to their “conveying” state. In such a way, by correctly setting the length of the conveyor, the constancy of the consumption of the time to failure with respect to variations in speed is assured. In the examined case, the length of the conveyor used to model the bounding machines was set equal to twice the length of the flow item. This allowed to model the variation of the speed from 200 to 400 soaps per minute in BM2.

4. EXPERIMENTS AND RESULTS

Simulation campaigns were set up to analyze variations on line throughput produced by changes in buffer size, capacity of BM2, and reliability of the bounding machines.

The settings of the machines are reported in the following. Reliability data for the machines were obtained by following the approach reported in Perrica, Fantuzzi, Grassi, Goldoni, and Raimondi (2008).

Stamper:

Rate: 1 rank per second.

Rank: 10 soaps.

Target production rate: 600 soaps/minute.

Machine target efficiency: 95%.

TTF: 5600 seconds mean, exponential distribution.

TTR: 180 seconds minimum, 300 seconds mean.

Gamma distribution with location 180, scale 240, shape 0.5, and maximum value of 1200 seconds.

Cartoner:

Machine target efficiency: 93.75%.

TTF: 1800 seconds mean, exponential distribution.

TTR: 30 seconds minimum, 120 seconds mean. Gamma

distribution with location 30, scale 112.5, shape 0.8, and maximum value of 1200 seconds.

Divider:

Machine target efficiency: 97.56%.

TTF: 3600 seconds mean, exponential distribution.

TTR: 15 seconds minimum, 90 seconds mean. Gamma

distribution with location 15, scale 93.75, shape 0.8, and maximum value of 1200 seconds.

Boudling machine 1:

Format: 6 soaps.

Target production rate: 430 soaps/minute.

Film change: manual.

Film running time: 1.2 hours (at 430 soaps/minute).

Mean film consumption per soap: 0.1395 film seconds per soap.

Mean film consumption per rank: 1.395 film seconds per rank.

Stop reasons:

1. Film change stop:
 - Mean time between two consecutive film changes: related to actual consumption.
 - Mean film change time: 60 seconds minimum, 300 seconds mean. Gamma distribution with location 60, scale 300, shape 0.8, and maximum value of 1200 seconds.
2. Jamming stops:
 - TTF: following values are considered: 1800, 2700, 3600, 4500, 5400 seconds. Exponential distribution.
 - TTR: 30 seconds minimum, 126 seconds mean. Gamma distribution with location 30, scale 120, shape 0.8, and maximum value of 1200 seconds.

Boudling machine 2:

Format: 2 soaps.

Target production rate: 200 soaps/minute (slow version), 400 soaps/minute (fast version).

Film change: automatic.

Film running time: 39 minutes (at 400 soaps/minute).

Mean film consumption per soap: 0.15 film seconds per soap.

Mean film consumption per rank: 1.5 film seconds per rank.

Stop reasons:

1. Film change stop:
 - The machine does not stop for film changing. It slows down to 100 soaps per minute for 2 seconds.
2. Jamming stops:
 - TTF: following values are considered: 1800, 2700, 3600, 4500, 5400 seconds. Exponential distribution.
 - TTR: 20 seconds minimum, 108 seconds mean. Gamma distribution with location 20, scale 135, shape 0.8, and maximum value of 1200 seconds.

Buffer:

Ideal operation without jamming, or in any case MTTF consistently higher than the one of other machines.

Two simulation campaigns were executed to assess the effects of buffer size, BM2 speed, and bounding machines reliability on line throughput. In the first simulation campaign the manual splice in BM1 was considered, whilst in the second campaign the automatic splice was introduced in BM1.

In both of the two campaigns the following settings were adopted.

Factors and levels

- Buffer size [ranks]: 4, 25, 50, 100, 150, 250, 350.
- BM2 speed [ranks/s]: 0.3334, 0.6667 (200 and 400 soaps/minute).
- BMs MTTF [s]: 1800, 2700, 3600, 4500, 5400.

Performance measure

Line asymptotic mean throughput, measured in soaps per minute.

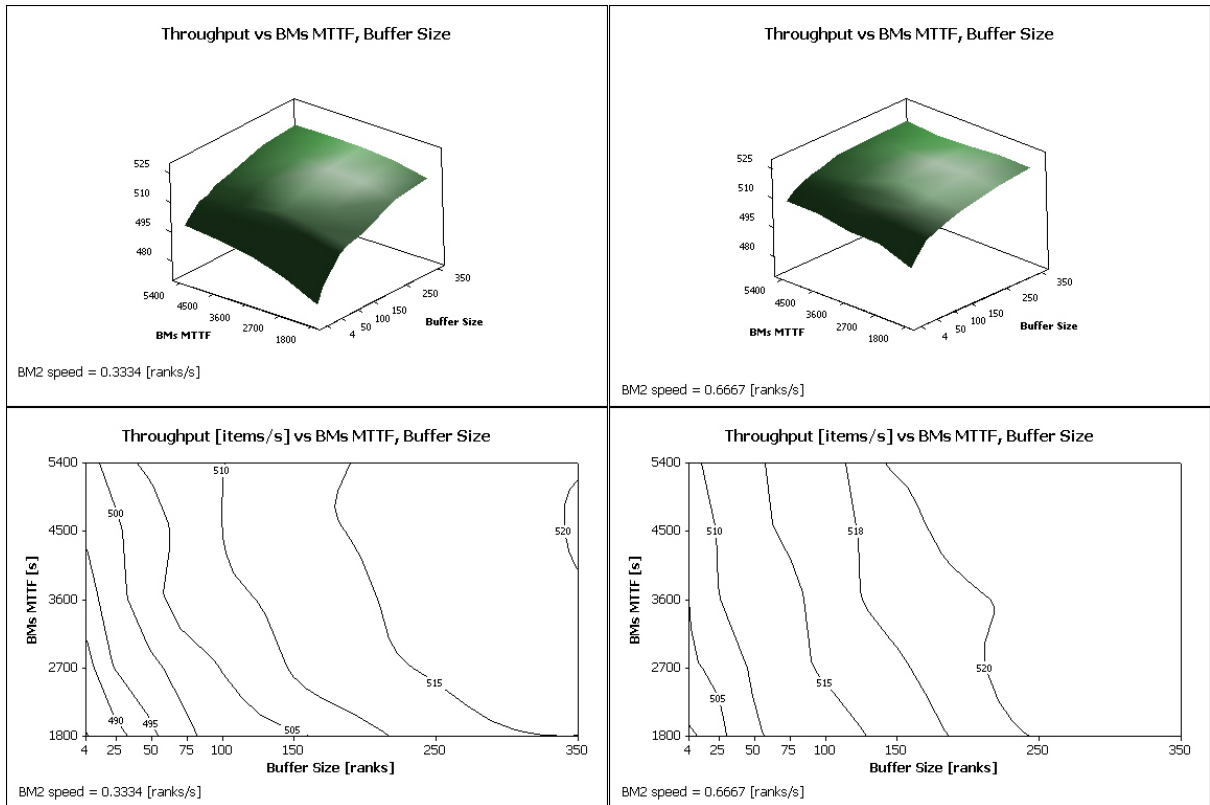
Divider control policy

Two ranks (20 soaps) on BM1 and 1 rank (10 soaps) on BM2, alternately. In the case in which BM2 speed is 400 soaps per minute, such a distribution is reversed each time the buffer reaches the full condition, and is restored as soon as the buffer becomes empty.

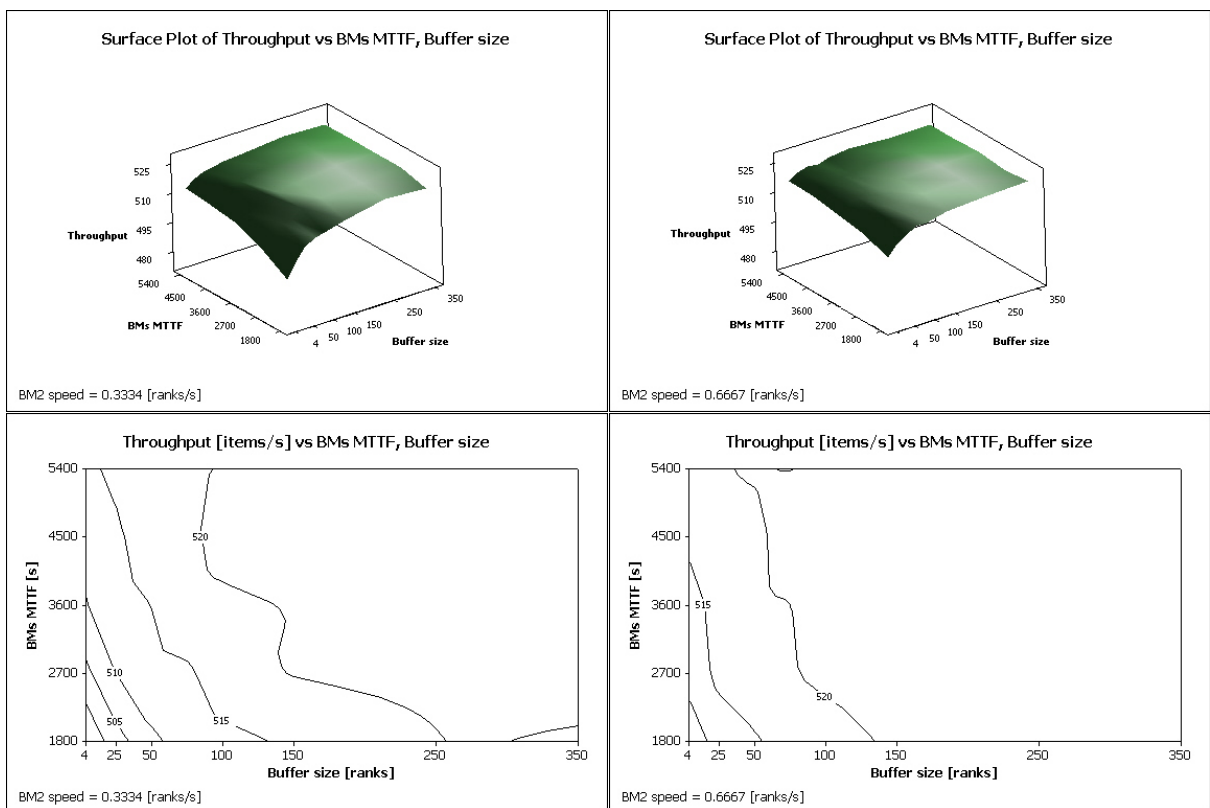
All on BM1 when BM2 is down, all on BM2 when buffer is full.

Experiments

- Approach: non-terminating system.
- Full factorial analysis.
- Number of repetitions for each combination of factors: 6.
- Total number of experiments: 480.



Figures 2a,b,c,d: Results with manual splice on BM1.



Figures 3a,b,c,d: Results with automatic splice on BM1.

- Running time of each experiment: 3,000,000 seconds (34.72 days).

The ANOVA test was executed to assess the statistical significance of the factors over the response (line throughput). In both of the two campaigns executed, the ANOVA stated that the three examined factors significantly affects the line throughput. When the manual splice is considered in BM1, the most significant factor is the speed of BM2, followed by the buffer size. The MTTF of the bundling machines is the less significant factor. On the contrary, when automatic splice is considered in BM1, the three factors assume an almost equal significance, stating that the manual film change operation on BM1 consistently affects the performance of the line and must be balanced by acting on buffer size or on downstream overcapacity (BM2 speed).

Figure 2 and Figure 3 report the surface plots and the contour plots of the line throughput over the three factors in the case in which manual splice or automatic splice is considered in BM1, respectively.

In both of the two simulation campaigns, the maximum line throughput that can be reached is about 525 soaps per minute, that corresponds to a line efficiency of about 87.5%. This states that, by adopting the highest levels of the factors examined, the negative effect of the manual splice on BM1 can be completely mitigated. Obviously, if the automatic splice is considered on BM1, values of the line productivity near the highest possible value can be reached without the need to adopt the highest levels of the factors.

By observing the contour plots reported in Figures 2 and 3 it can be pointed out that the adoption of a BM2 with a speed of 400 soaps per minute, corresponding to a total nominal capacity of the downstream of 830 soaps per minute, makes it possible to almost completely absorb the negative effects of the manual splice. Moreover, it can be noticed that such an increase in nominal capacity of the BM2 allows to obtain a consistent line throughput without the need to introduce a buffer (consider that a buffer of 4 ranks corresponds to the accumulation that can be obtained on the conveyor that connects the divider with the machine). This is also a consequence of the fact that, having packaging lines a high production rate, only a large buffer can provide the right time coverage to compensate the average time needed to restore the machine after jamming occurrence. Another interesting aspect related to the adoption of an overcapacity in the downstream is that the curves representing constant line throughput tend to assume a vertical trend, thus implying that variations in BMs reliability have a less effect on line performance. This is important since bundling machines can undergo variations in MTTF as the format they are configured to produce changes.

Finally, it can be observed that, in each of the examined cases, an increase in BMs MTTF over the value of 3600 seconds produces limited effects on line throughput. On the contrary, it is most important to

avoid a reduction of BMs MTTF under the value of 3600 seconds, because in such a situation a significant loss in line performance is emphasized, mainly when overcapacity is not present in the downstream.

5. CONCLUSION

Performance analysis of production lines characterized by the capability to produce different product formats at the same time are addressed in this paper.

Starting from an actual case study regarding a soap packaging line, crucial aspects influencing line performance are investigated and the effects of the most important key factors are studied. Guidelines to support the design phase for these specific kind of lines are then derived and provided to the reader.

REFERENCES

- Alvarez-Vargas, R., Dallery, Y., and David, R. (1994). A study of the continuous flow model of production lines with unreliable machines and finite buffers. *Journal of Manufacturing Systems*, 13, 221–234.
- Curcio, D., Longo, F., and Mirabelli, G. (2007). Manufacturing process management using a flexible modeling and simulation approach. *Proceedings of the 2007 Winter Simulation Conference*, 1594–1600.
- Dallery, Y., David, R., and Xie, X. (1989). Approximate analysis of transfer lines with unreliable machines and finite buffers. *IEEE Transactions on Automatic Control*, 34, 943–953.
- Dallery, Y. and Gershwin, S. (1992). Manufacturing flow line systems: a review of models and analytical results. *Queueing Systems*, 12, 3–94.
- David, R., Xie, X., and Dallery, Y. (1990). Properties of continuous models of transfer lines with unreliable machines and finite buffers. *IMA Journal of Management Mathematics*, 2, 281–308.
- Diamantidis, A. and Papadopoulos, C. (2006). Markovian analysis of a discrete material manufacturing system with merge operations, operation-dependent and idleness failures. *Computers & Industrial Engineering*, 50, 466–487.
- Fu, M. and Xie, X. (2002). Derivative estimation for buffer capacity of continuous transfer lines subject to operation-dependent failures. *Discrete Event Dynamic Systems*, 12, 447–469.
- Gershwin, S. and Burman, M. (2000). A decomposition method for analyzing inhomogeneous assembly/disassembly systems. *Annals of Operations Research*, 93, 91–115.
- Gujarathi, N., Ogale, R., and Gupta, T. (2004). Production capacity analysis of a shock absorber assembly line using simulation. *Proceedings of the 2004 Winter Simulation Conference*, 1213–1217.
- Harrell, C. and Gladwin, B. (2007). Productivity improvement in appliance manufacturing. *Proceedings of the 2007 Winter Simulation Conference*, 1610–1614.
- Patel, V., Ashby, J., and Ma, J. (2002). Discrete event simulation in automotive final process system. *Proceedings of the 2002 Winter Simulation Conference*, 1030–1034.
- Perrica, G., Fantuzzi, C., Grassi, A., Goldoni, G., and Raimondi, F. (2008). Time to failure and time to repair profiles identification. *Proceedings of the 5th International Conference on Simulation and Modelling in the Food and Bio-Industry – FOODSIM 2008*, 59–64.

SIMULATION AS SUPPORT FOR LAYOUT REDESIGN AND PRODUCTION LINE MOVING

Marina Massei^(a), Francesco Longo^(b), Federico Tarone^(b), Alberto Tremori^(c)

^(a)MISS Genoa, DIPTM University of Genoa

^(b)SIMULATION TEAM

^(c)MAST

^(a)massei@itim.unige.it

^(b){francesco.longo, federico.tarone}@simulationteam.com

^(c)alberto.tremori@mastsrl.eu

ABSTRACT

The authors present an innovative approach to facility layout re-organization as well as industrial facility moving based on the use of simulation and intelligent optimization techniques. This methodology has been applied in real cases and in particular for design of new production plants. The main advantages in the combined use of simulation and intelligent optimizations are the possibility to analyze and measure solutions proposed in advance, considering production flows and targets, physical constrains, material paths, etc.

Keywords: Simulation, Optimization, Layout Engineering, Production Relocation

1. INTRODUCTION

Design or re-engineering of a new facility is a very complex problem with big investments for the company. It is necessary to consider several variables: huge amount of historical data, but, often, uncertain, from different sources and affected by stochastic behaviors; furthermore such information must be combined to understand the real behavior of the facility. Simulation and optimization are solution traditionally adopted in this kind of analysis and that can be adopted at the same time with a dynamic interaction between simulator and optimization algorithms. The interaction allows evaluating the performances of proposed solutions.

In this paper the authors present an approach derived from these cases. The methodology adopted combines different models (shop-floor, machineries production flows) with an intelligent optimizers module based on Genetic Algorithms (GAs). The results are displayed by the use of a three dimensional Virtual Environment.

The goal of the projects in which these researches are applied are in fact often different: the optimization of the plant layout of course, but also a tool to support the movement from the old to the new building, to allow the production managers to study new layouts based on new market scenarios. This paper describes the methodology adopted and the solution developed to support companies in the re-engineering of the

production lay-out. Methodology and Solution are named LEXIS (Layout Excellence & Integrated Solutions). LEXIS is divided in different phases (data collection, model design, 3D visualization module ...). Software is developed in C++ (simulator and GA optimization).

The paper presents in particular the application of LEXIS for supporting the development of a new Production Facility for an Aerospace Industry where this simulation model was extensively used in connection with genetic algorithms optimization.

2. THE INDUSTRIAL SCENARIO

The scenario in which LEXIS has been applied is the design of a new productive site and planning of movement from one of the old plants. Actually the company production is based on two plants one for manufacturing of planes parts and engines and the other for plane assembly. The manufacturing is very old and presents a lot of constrains in terms of space and internal mobility. So the company management decided to start the design of a new production plant and to close the old area. The new plant has to guarantee the flexibility that new market scenarios ask.



Figure 1: the plant

The Company production facilities are located in the northwest of Italy. The new state-of-the-art facility will be located to about 30km to the old one in a wide

open area and will be designed to implement the latest lean manufacturing technology. The new lean-production approach that will be adopted in the new plant will increase production capacity, efficiency and optimize workflow. These are are imporvement that has to be followed by the Company in response to the rapid expansion of its business. Furthermore this will help the Company to successfully meet the challenges of the market in term of flexibility and produciton flows, time to market, ecc...

The scope of this project, as already mentioned before, is to support the Company to define the new plant layout and to plan the move from the old to the new with the minumum impact on productive flows.

The project is divided into different main phases: Data Collection and Certification, Production Flow Definition, Model Development, Optimization Algorithms Development, Integration with the 3D Synthetic Environment, Support and Consultancy for Layout Optimization. At the beginning of the study, the authors were requested to evaluate different implementation solutions in order to guarantee the users with the capability to maintain the simulation implementation. Different COTS (Commercial Off-the-Shelves) have been evaluated in term of their technical and functional aspects:

Layout Optimization Capabilities, Graphics Capabilities, Compatibility with other Software, Maintainability & Stability, Acquisition, Service and Training Procedures. The actual productive status of the plant is based on a set of over 400 machineries. Every machinery is served by several small local warehouse areas with raw material, spare parts and finite goods. Also there are areas for work in progress (WIP), architectural elements, ancillary entities and services.

3. LEXIS ARCHITECTURE

The general architecture of the system integrates the mentioned modules: simulation model, GA optimization module and 3D virtual environment for visualization. The Genetic Algorithms otpimization module provides solution (both local and global) to the layout simulator. This paper is focused mainly on the description of how GA has been developed and integrated. Taking into consideration the variables which have to be optimized, the Genetic Algorithms allow the researchers to use the final user suggestions as seed for the whole optimization process.

4. CONCEPTUAL MODEL BY THE OODA

The conceptual model has been developed by the OODA (Object Oriented Design and Analysis) approach.

According to this here below are listed the objects used to represent the plant layout form evry point of view:

Layout Object	IdLO		Layout Object (LO) ID
	SLO		LO Status
	aLO		Orientation
	DxLO, DyLO, DzLO	Delta Position from	
	i) Department Reference Point		
	FnxLO, FnyLO, FnzLO	Foundation Size (X,Y,Z)	
	FnmLO		Foundation Mass
	FnsLO		Foundation Surface
	LxLO, LyLO, LzLO	LO Size (X,Y,Z)	
	LaLO		LO Area
	PDI/ACILO	Production Department Id	
	i) and Activity Cost ID		
	DescriptionLO	LO Description	
	CodeLO		LO Code
	ConstructorLO	LO Constructor	
	ModelLO		LO Model
	TypeLO		LO Type
	DepartmentLO	LO Department	
	SmokeLO		Smoke Evacuation Requirements
	PrLO		Productivity
	MobLO		Mobility
	NoiseLO		Noise Generation Level
	SensLO		Sensibility Grade respect Noise
	SOLO		Operating Status
	DispLO		Availability starting date (new devices)
	LineLO		Production Line
	TimeLO		Expected Time required for moving from old to new facility

Service Box	idSB		Service Box(SB) Code
	loSB		Associated Layout Object
	dxSB,dySB, dzSB	Distances from LO Reference Point	
	lxSB,lySB,lz SB	SB Size	
	b		SB Orientation

Flows	idf		Flow Code
	Typef		Flow Type
	Sf		Description
	From_Wheref	SB Origin	
	To_Wheref	SB Destination	
	MainProductIDf	Main Product ID	
	Items/MainProductf	Production Ratio	
	ItemIDf		Item ID
	Codef		Item Code
	Tf1		Set Up Time
	Tf2		Unit Production Time
	LTf		Production Lot

Products	idp		Final Product ID
	Qp		Planned Quantity
	Codep		Final Product Code

Items	idi		Item ID
	Si		Item Description
	Sxi, Sxi, Sxi,	Item Size (X,Y,Z)	
	Ni		Number in Packaging Unit
	MMi		Movement Mode
	Codei		Item Code

Corridor	idc		Corridor ID
	Typec		Corridor Type
	Descriptionc	Corridor Description	
	From_Wherec	Starting Node	
	To_Wherec	Ending Node	
	Sizec		Corridor Size

Corridor Node	idcn		Corridor Node ID
	Typecn		Corridor Node Type
	Descriptioncn	Corridor Node Description	
	Xcn, Ycn, Zcn	Corridor Node Location	

Handling Device	Idhd		Handling Device ID
	Typehd		Handling Device Type
	Descriptionhd	Handling Device Description	
	Caphd		Handling Device Capability
	Coshd		Handling Device Cost
	Speedhd		Handling Device Speed
	Nhd		Handling Device Number

Production Department	ID Dp		LO Department
	SDp		Department Description
	DphDp		Department Orientation
	Dpx Dp, Dpy Dp, Dpz Dp	Department Location (X,Y,Z)	

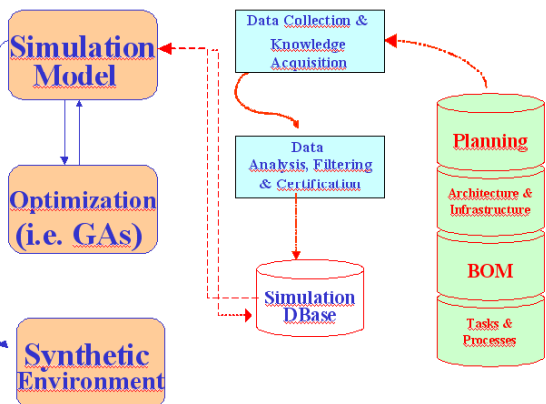


Figure 2: LEXIS Architecture

The main object used to create the model are the Layout Objects. With this kind of object have been represented machineries, warehouses, architectural elements, offices, wardrobes etc. A layout object can be associated with one or more Service Boxes. Service Box

are used to represent all the service area that serve a Layout Object: for instance every machinery needs area for loading and un-loading or place to store temporarily materials before the pick up.

Layout Objects and Service Boxes are represented in the 3D visualization system but the z axes are considered also in the 2D optimization environment: the optimization, in fact, can take in count as a physical constrains the height of an object.

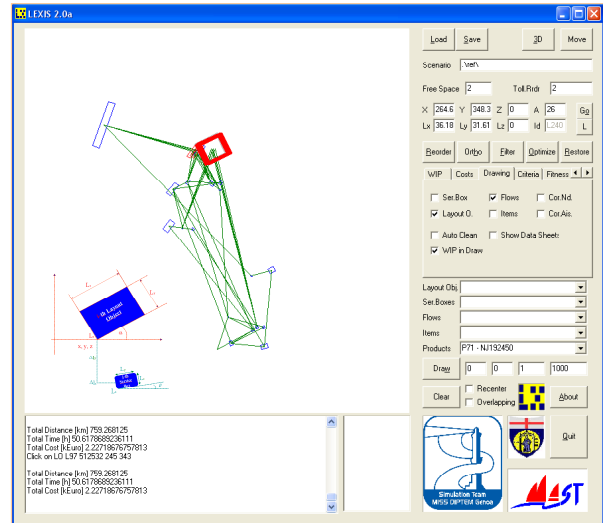


Figure 3: LO and SB

Flows connect two Service Boxes. It is possible to associate different Items to a Flow (raw materials, components, intermediate products, finished goods). Handling Devices are used to represent internal transportation of Items over a Flow while the Item quantities are attributed to each Flow in relation to the final product quantities, so all the quantities and costs are easily updated based on production mix expectations. Flows in the Layout, involving two different final products (A & B) are provided in the following picture.

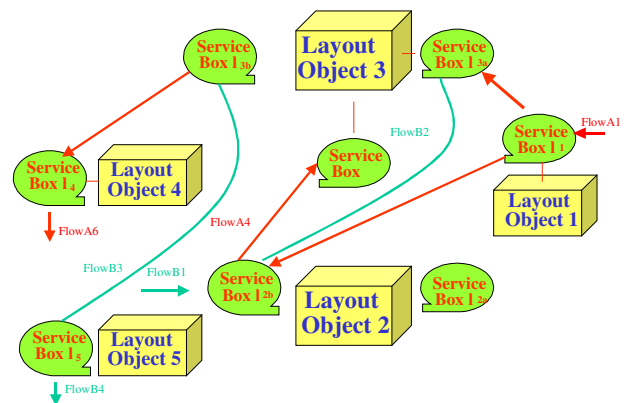


Figure 4: Flows Scheme

5. METHODOLOGY for DATA COLLECTION

As mentioned in the introduction one of the phase of the described project is the Data Collection and Certification. This is of course a crucial phase due to the huge amount of information necessary to feed the model with the critical problem of uncertainty of data. We can briefly summarize as follow what kind of information has to be collected in this kind of project

- Technical details about machineries (dimensions, productivity, foundations, operative range, tolerances)
- Technical drawing
- Production Process and Flows (global and for every machinery)

These data contribute to define the machines into the simulator, making possible to evaluate the volumes for the related buffers and to define the optimum layout to facilitate the loading and unloading of materials. An active LO is each element hosting a task or production process, while the passive elements are entities representing architecture component, infrastructures or ancillary objects.

5.1 Scenario definition

To understand the consequences that different hypotheses can generate it is important to understand and define the scenario. In the project described for instance it has been crucial consider and understand the Company market evolution in terms of demamand and consequently in terms of new products or new flows. According to this the layout proposed obviously need to be robust.

5.2 Scenario definition

Form the very beginning of the project it has been clear that data collection has been very complicated. Infact the reserch team ought to work in strong connection with different Company Directorates to collect information: industrial engineering, processes, production, maintenance, etc. It has been also clear thata there was a lack in connection among different source of data. The authors in this kind of project ususally started to collect:

- Bill of Material (BOM)
- Machine Parameters & Drawings
- Production Process Definition & Task List

In the described project the information related to the processes, but with no direct “link” to the machines, are not structured. Furthermore the Company produces two main line of products (engines and aircrafts) and related data are structured in different ways with different formats. Due to this it became necessary to combine the Production Processes with BOMs. In this way it was possible to define the sequential procedures

for finished goods as well as the structure of the initial, intermediate and final warehouses.

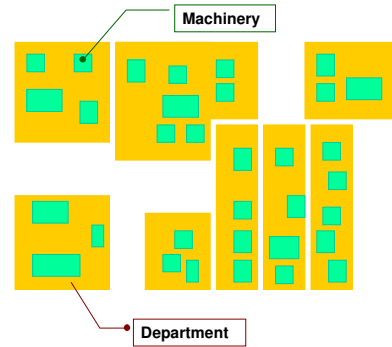


Figure 5: Layout Re-engineering Multilevel approach

Another problem is due to the fact that production tasks can be carried out in the plant under examination (today the old one, and in the future the new one under costruction) can be moved to other production site or can be outsourced: it is necessary to extract the items related to the layout taking into account the task list. At the same time the model has been developed flexible to consider the different possibilites (production inside or outside the plant)

6. SIMULATION AND BOM: BILL OF MATERIAL

To associate production flows to item it is fundamental to extract items by Decomposition of the main products in term of elements. The BOM in the described project is characterized by 14 different levels form final product to sub-components.

6.1 Machinery Lists & Task

Two identifiers has been defined for each element required by the Task List:

- Production Department Id (PDI)
- Activity Cost ID (ACI)
-

The first input source has been the maintenance dept. DB; this includes description for every machinery in the plant. This list has been completed with all the areas required by the tasks and processes, this based on joint review with subject matter experts of production process definition and task list. The final result of this activity it has been possible to complete a full list of all Layout Objects in the plant. However the Machine DB, developed by maintenance dept., was mostly intendende to support the move from the old to the new facility; according to this it was structured considering new machineries and existing elements to be moved. In this way, with no further actions, it was clear that the machine lists could be inconsistent if compare with the Production Department Id and Activity Cost ID. It is interesting to outline the presence of “exceptions” in the management of this data. The authors had to normalize

these exceptions to be included in the model. The following figure proposes an example of the current situation with related definition of elements, this based on its production department. In order to combine the two data set in a single consistent element, it was necessary to avoid duplications and identify connection at the same time. The authors worked with Subject Matter Experts (SME) in order to define proper details for modeling the active elements representing Layout Objects. At the end of this critical phase it was possible to complete the list of Layout Objects that have to be moved to the new facility, including new one to be bought and all that objects, not machineries, but areas or other objects critical in the plant layout such as testing area or working banks. In this way it has been possible to complete a list of active objects operating in some way in the layout. The authors decided also to collect pictures for each machine, to create more realistic views of the 3D objects.

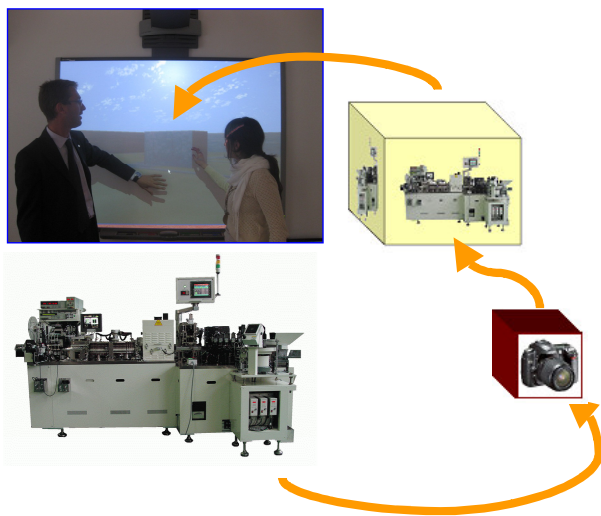


Figure 6: 3D representation by collection of images

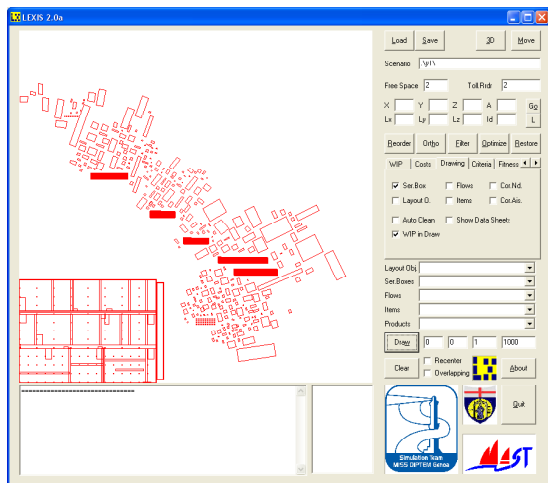


Figure 7: LEXIS Interface and Layout reorganization

6.2 Old Plant Layout

The research team analyze the current layout to understand and measure its effectiveness as reference for

comparison with the proposed new plant solutions. Using the technical drawing it has been possible to measure the current position of each machine in relative coordinates respect its production dept..size and position of each production dept. itself and warehouses and other departments. Starting from the current situation has allowed to a very simple but easy to be obtained optimization as a starting point and in any case as a reference for any new solution.

6.3 Architectural Information

For creating the layout model it has been necessary to collect all the architectural information about the new building:

- dimensions
- foundations
- material structure,
- position of the columns,
- position of the toilets, wardrobes and offices,
- special plants
- emergency exits

So the next step has been to complete the layout objects from architectural elements of the future factory

7. CONCLUSION

This paper provides an overview on an integrated approach for layout engineering; therefore currently in most case these problems require strong interactions among decision makers as well as proper evaluation of relocation alternatives or production moving solutions; LEXIS based on its object-oriented nature allows very flexible multilevel analysis of the problem; the integration with intelligent optimizer provides an additional advantage in facing complex problem, both for preliminary screening of alternative and for finalizing optimal solutions; last, but not least, the virtual reality integration represent an interesting opportunity to proceed interactively with decision makers in evaluating each single open issue. Currently the authors are working both in introducing new features in the optimization algorithms and in developing new projects for real industries as opportunity to the further enhance the models.

ACKNOWLEDGMENTS

The authors would like to thank Marta Caneparo, Giorgio Garassino, Enrigo Musso, and Adeline Queau for their support in to the scenario definition for the present study.

REFERENCES

- Abdelghani S. 1995. "Simulated Annealing for Manufacturing Systems Layout Design", European Journal of Operational Research, 82
- Askin R. G. and Standridge C. 1993. "Modeling and Analysis of Manufacturing Systems", Wiley, NYC

- Bruzzone A.G. 1998. "Stochastic Planning & Scheduling Production For Moulding Processes Using Simulation", Proc. of SCSC, Reno, July
- Bruzzone A.G. and Giribone P. 1998. "Simulating Assembly Processes in Automotive Support Industries for Production and Design Improvement", Proc.of XV Simulator International, Boston,
- Bruzzone A.G. and Giribone P. 1998. "Robust and Central Composite Design as Collaborative Techniques for Production Planning using Simulation", Proc.of Eurosim98, Helsinki
- Bruzzone A.G. and Giribone P. 1999. "Industrial Engineering Tools & Techniques for Development of New Manufacturing Processes in Automotive Parts Production", Proc. of Summer Computer Simulation Conference, Chicago, July 17-21
- Bruzzone A.G., Giribone P. and Vio F. 1999. "Genetic Algorithms and Simulation for Supporting Layout Re-Engineering of Automotive Component Production Facilities", Int.Journal of Flexible Automation and Intelligent Manufacturing, Vol 7, Issue 3-4, 379-391 ISSN 1064-3645
- Bruzzone A.G. and Mosca R. 2002. "Simulation And Fuzzy Logic Decision Support System As An Integrated Approach For Distributed Planning Of Production", Proc.of FAIM2002, Dresden, July
- Bruzzone A.G. and Saavedra S. 2005. "Production Optimization Of A Flexible Manufacturing Cell Using Genetic Algorithms And Simulation", Proceedings of MAS2005, Bergoggi, October 16-18
- Bruzzone A.G., Tremori A., Massei M., Caneparo M., 2009 "Modelling Layout for Integrating Simulation, Optimization and Virtual Reality" Proc. of ECMS 2009, Madrid, SP
- Bruzzone A.G. and Vio F. 1999. "MACS: A Framework Integrating Genetic Algorithms and Simulation for Supporting Layout Re-engineering", Proceedings of HMS99, Genoa, September 16-18
- Bruzzone Agostino G. 1994. "Potential of Reduced Experimental Designs in Object-Oriented Modeling of a Manufacturing Department for Film Production", Proceedings of ESM94, Barcelona, SP
- De Luca, P. 2003. "Simulazione e Ottimizzazione per il progetto di un impianto manifatturiero a celle di lavorazione: un approccio integrato", Thesis, Università di Roma "Tor Vergata", Italy,
- Francis R.L., McGinnis F., White J.A. 1991. "Facility Layout and Location: An Analytical Approach", Prentice-Hall, NYC
- Giribone P. and Bruzzone A.G. 1995. "Artificial Neural Networks as a System for Production Planning in an Alimentary Plant", Proc. of Simulators International XII, Phoenix, April 9-13
- Giribone P. and Bruzzone A.G. 1997. "Production Games: An Improvement in the Education and Training for Engineers working in Production and Logistics", Proc.of European Simulation Symposium, Passau, Germany, October 20-23
- Glover F., Kelly J.P. and Laguna M. 1999. "New Advances for wedding optimization and simulation" Proceedings of the 1999 Winter Simulation Conference, Phoenix, AZ
- Heragu S.S. 2008. "Facility Design", CRC, NYC
- Kirkpatrick, S, Gelatt C.D. and Vecchi C. 1983 "Optimization by Simulated Annealing", 1983, Science 220, 4598, 671-680
- Malakooti B. and Ziyong Y. 2002, "Multiple criteria approach and generation of efficient alternatives form machine-part family formation in group technology", IIE Transactions, 34, 837-846.
- Mohsen M.D. Hassan, "Layout Design in Group Technology Manufacturing", International Journal of Production Economics, 1995, 38, 173-188.
- Mosca R., Giribone P. and Bruzzone A.G. 1996 "Study of the Behaviour of Flexible Production Systems using Neural Nets", International Journal of Production & Control, vol. 7, no.5, 462-470
- Mosca R., P.Giribone and A.G. Bruzzone 1997 "Information Technology and Artificial Intelligence to Support Production Management in Industrial Plants", Proceedings of AI'97, Innsbruck, Austria
- Mosca R., P.Giribone, A.G.Bruzzone, A.Orsoni and S.Sadowski. 1997 "Evaluation and Analysis by Simulation of a Production Line Model Built with Back-Propagation Neural Networks", Int.Journal of Modelling and Simulation, Vol.17, no.2, pp.72-77
- Phillips E. J. 1997. "Manufacturing Plant Layout: Fundamentals and Fine Points of Optimum Facility Design", Society of Manufacturing Engineers
- Santos J., Wysk R. and Torres J.M. 2006 "Improving Production with Lean Thinking", Wiley & Sons, NYC
- Selim H.M., Askin R.G. and Vakharia A.J. 1983 "Cell formation in Group Technology: review, evaluation and directions for future research", Computers Ind. Eng., 34, N°1, 3-20.
- Thomas P. J. 1999. "Simulation of Industrial Processes for Control Engineers", Butterworth-Heinemann
- Yong Y. 2002 "Approach to Solving Part Family/Machine Group Formation Problem in Cellular Manufacturing", Ph.D. dissertation in Industrial and Management Science, Graduate School of Economics and Management, Tohoku University, Japan

AUTHORS BIOGRAPHY

Marina Massei is currently enrolled in DIPTM University of Genoa as member of the Simulation Team of Prof.Agostino Bruzzone. She is involved in the organization of International Scientific Events (i.e. Summer Computer Simulation Conference 2003-Montreal, 2004-San Jose', 2005-Philadelphia) She

served as Chair for Applications in Management, Planning & Forecasting Workshop in SummerSim 2004, 2005, 2006. She is Associate Director of the MISS Center at the University of Perugia She was Finance Control & Administration Director of MAST.

Francesco Longo was born in Crotona on February the 8th 1979. He took his degree in Mechanical Engineering, summa cum Laude, in October 2002 from the University of Calabria. He received his Ph.D. in Mechanical Engineering from the Department of Mechanical Engineering at University of Calabria in January 2006. He is currently researcher in the same Department. His research interests include Modeling & Simulation for production systems design and supply chain management. He was General Chairman of the 20th European Modeling & Simulation Symposium He was Program Co-Chair of the 4th International Modelling Mediterranean Multiconference and Program Co-Chair of the 19th European Modeling & Simulation Symposium. He belongs to the International Program Committee (IPC) of the Summer Computer Simulation Conference and Track Chair of the Emergency Simulation session in the same conference. He belongs to the IPC of the European Conference on Modeling & Simulation. He works as reviewer for the most important conferences on Modeling & Simulation such as the Summer Computer Simulation Conference, the European Conference on Modeling & Simulation, the Winter Simulation Conference, the Workshop on Applied Simulation, etc. He works also as reviewer for the international journal Computer & Industrial Engineering. He is Responsible of the Modeling & Simulation Center – Laboratory of Enterprise Solutions (MSC-LES). He teaches “Design of Production Systems” and “Laboratory of Industrial Plants” respectively for students of the 2nd and 5th year in Management Engineering. He is also tutor on the courses “Industrial Plants” and “Production Systems Management” respectively for students of the 3rd and 5th year in Mechanical Engineering.

Alberto Tremori is member of the Simulation Team at DIPTTEM University of Genoa and Liophant Simulation. His expertise is simulators development for supporting process re-engineering, logistics, transportation systems and warehouse automation. From 1997 he was involved in several Simulation Projects (i.e joint project involving National Simulation Center, Institute for Simulation and Training, Kennedy Space Center, McLeod Institute of Simulation Science). He currently serves as President of MAST.

Federico Tarone is Phd student at DIPTTEM, University of Genoa. He is currently involved in some Simulation projects in the field of Logistics, Inventory Management and Web Applications. He is member of the Liophant Simulation. In 2008 he attended the I3M International Conference as speaker.

SIMULATION, RISKS MODELING AND SENSORS TECHNOLOGIES FOR CONTAINER TERMINALS SECURITY

Francesco Longo^(a), Giovanni Mirabelli^(a), Alberto Tremori^(b)

^(a)Modeling & Simulation Center Laboratory of Enterprise Solutions (MSC-LES)
M&SNet (McLeod Modeling & Simulation Network)
University of Calabria, Italy
f.longo@unical.it, g.mirabelli@unical.it – URL www.msc-les.org

^(b)Simulation Team
Via Molinero 1, 17100 Savona, Italy
Alberto.tremori@simulationteam.com – URL www.simulationteam.com

ABSTRACT

The goal of this paper is to present a research program just started at MSC-LES (Modeling & Simulation Center Laboratory of Enterprise Solutions, MSC-LES, at University of Calabria). The SEAPORTS research program presents an advanced approach for investigating and analyzing the security problem within marine ports environment in order to find out security gaps and proceed with the redesign of the most critical security procedures and infrastructures. SEAPORTS also investigates the effects of port security procedures on the global supply chain.

Keywords: Containers Terminals, Risks Modeling, Sensors Technologies, Simulation

1. INTRODUCTION

The SEAPORTS approach is based on the idea to consider the whole complexity of a real marine port. Marine ports facilities are characterized by complex interactions among numerous stochastic factors and variables. The complexity of marine ports also depends on technical aspects characterizing port facilities (i.e. cranes operations, handling equipment movements, equipment scheduling, etc.). To take into account such complexity, SEAPORTS uses a Modeling & Simulation Infrastructure (MSI) as fundamental part of its architecture, for recreating, in a synthetic environment, all the marine port main operations. The Modeling & Simulation Infrastructure will be developed on the basis of a real marine port scenario (called port reference scenario).

A real marine port has several vulnerability points – such as port environment extension, access not adequately monitored, no complete containers inspections, entrance allowed to unauthorized persons and it is subjected to threats as thefts, smuggling, terrorism, vandalism and so on. On the other side a marine port adopts sensors technologies, security procedures and security plans devoted to

reduce port vulnerabilities, increase protection capability and security. An approach devoted to support the review and redesign of security procedures and infrastructures (eventually identifying security gaps), in addition to the main port operations, must also recreate risks and threats and the security procedures and sensors technologies to detect them. To this end SEAPORTS propose to develop two platforms respectively called *Key Risks Modelling Platform* (KRMP) and *Sensors Technology Management Platform* (STMP). The first one is a generator of virtual risks and threats (generator of virtual risk scenarios), the second one recreates virtual sensors technologies for detecting risks and threats within the marine port.

The whole SEAPORTS architecture, based on the System of systems idea, is the integration of the three architectural components above mentioned: the MSI, the KRMP and the STMP. In effect those three elements recreate, in a synthetic environment, the whole port reference scenario in terms of virtual port main operations, virtual threats and risks generation, threats and risks virtual detection.

The KRMP will affect the main port operations (generated by the MSI) at least by generating two virtual risk scenarios. Note that the SEAPORTS research team has already defined the virtual risk scenarios in terms of threats, however such scenarios could be modified or additional virtual risk scenarios could be added on the basis of data collected in the port reference scenario. Similarly the STMP will initially recreate the sensors technologies available in the port reference scenario, then additional sensors technologies will be added.

The way to detect threats within a marine port environment by using available sensors technologies depends on security procedures used. Security procedures are suggested by normative and standards and by current best practices. A research effort will be carried out in order to identify security procedures within the port reference scenario: such security procedures will be the logical relation-

ships for integrating correctly the virtual sensors (STMP) within the virtual port operations (MSI).

Finally a case study based on the port reference scenario will be developed and SEAPORTS will be used for supporting the review, test and certification of the security procedures currently used, for security gaps identification, for design and re-engineering of security procedures/infrastructure (on the basis of gaps analysis, additional virtual risk scenarios and additional sensors technologies) and for the evaluation of effects of security advances in marine ports on the global supply chain.

2. MARINE PORTS SECURITY: A STATE OF THE ART OVERVIEW

Finding ways to intercept illicit materials shipped via the maritime transportation system, detecting risks within the marine port are exceedingly difficult tasks. Practical complications involve the design of security procedures as well as the impact on shipping and import port facilities and the tradeoffs between costs and potential risks, among others. Starting from 2001, the first step toward the accomplishment of security measures inside marine ports was the updating and development of normative and standards. The most important normative proposed or revised after 9/11 include the following:

- Container Security Initiative (CSI);
- International Convention for the Safety of Life at Sea (SOLAS);
- Maritime Transportation Security Act of 2002 (MTSA);
- Convention for the Suppression of Unlawful Acts (SUA).

Specific details on normative and standards can be found in the Marine Log (2004-a, 2004-b), Safety at Sea International (2003) and Security Management (2003). Note that normative and standard provides support in terms of security procedures/plans but they do not directly deal with methodologies, tools, equipment and technological advances to secure marine ports and do not evaluate the impact of security advances on the overall efficiency of a marine port.

Note that, in the field of marine ports, the research works proposed in the past have always considered the port operations efficiency rather than security concerns. Typical marine ports operations include cargo flows, berth management, yard management, etc. A number of stochastic factors and variables interact and evolve during the execution of port operations. The complexity of a marine port scenario requires to apply ad-hoc models (in fact, analytical approaches rarely succeed in properly identifying optimal solutions). In designing and managing such kind of facilities, it is becoming evident, the importance of using an approach capable of considering the system being studied

as a whole. To this end, the Modelling & Simulation (M&S) approach is usually the most suitable methodology for studying marine ports behaviour, conducting what-if analysis and defining technical configurations and management policies to be applied. In effect, the state of the art overview (reported below) highlights that: (i) the M&S approach is very effective for design and management of marine port operations; (ii) design and management of marine port operations still provide challenging problems to researchers working in this field.

Simulation is often used for studying the behaviour and the performance of marine terminals. Shabayek and Yeung (2002) investigate to what extent a simulation model could predict the actual container terminal operations with a high order of accuracy. Kia et al (2002) use the simulation for comparing two different container terminal scenarios in relation to handling equipment and their impact on the capacity of the terminal. Furthermore, simulation is often used for carrying out optimizations on specific scheduling and resources allocation problems (Gambardella et al., 2001). In addition, in other research works simulation is used for port design and as a decision support tool (Moorthy and Teo, 2006; Ottjes et al., 2006).

Until 2001 the research works did not consider the marine terminal security. The 9/11 has strongly underlined the need of designing (review, test and certificate) and integrating the security procedures within the marine port operations (in effect, marine ports security is a “young” research field). Consider the containers inspection operations devoted to detect illicit materials sited within containers: one of the main issues is the trade-off between the percentage of containers to be inspected and the consequent delays of departure ships, trucks and trains. Lewis et al. (2002) propose an approach (based on a best-first heuristic) for evaluating the balance between the percentage of containers to be inspected and the delays of the departure vessels (even if formulation being proposed is quite simple and it is not capable of recreating with satisfactory accuracy a real container terminal scenario). Babul (2004) and Wenk (2004) respectively propose a detailed analysis of the measures being undertaken for ports vulnerabilities a list of security precautions to be adopted within container terminals. Further research studies have been proposed on technologies for containers inspection (Brown, 2003; Orphan et al., 2005).

From 2005, authors are continuously carrying out research activities on marine ports security and effects on the global supply chain, investigating different problems. Longo & Bruzzone (2005-a), by using simulation, show that the increase of physical containers inspections beyond a single digit percentage brings main port operations to a halt. Longo et al. (2005-b), Longo et al. (2005-c) and Longo et al. (2006) investigates the integration of containers inspections operations within the normal port operations by combining simulation, design of experiments and

analysis of variance. Bruzzone, Longo and Papoff (2005) and Bocca et al. (2005) present global metrics for logistic facilities security (including marine terminals). Specific research works have been also developed on the effects of security procedures on the global supply chain, by considering the impact of catastrophic events (Longo, 2007), the capability of supply chain of assuring continuity and resilience (Longo and Oren, 2008; Bruzzone et al., 2006) also considering advanced approach based on emergence, anticipation and multisimulation (Oren and Longo, 2008).

As stated in the project objectives, the main contribution of SEAPORTS to the state of the art overview is to enhance protection capability and security within marine ports by supporting: (i) review, test and certification of ports security procedures; (ii) security gaps identification; (iii) design and re-engineering of security procedures/infrastructures; (iv) evaluation of effects of security advances in marine ports on the global supply chain.

The state of the art overview highlights that there is a lack of research studies on tools for supporting security procedures enhancement within marine ports. As far as we know most of the research projects, also those supported by the European Union on security issues and crisis management within marine ports, look in different directions: they propose advanced architectures for surveillance in wide or small maritime areas (addressing problems such as airborne surveillance) or provide unified data for improving international cooperation on marine ports security. On going research projects do not tackle the security problem within a marine terminal from the point of view of review and certification of security procedures, gaps analysis, design of new security procedures and effects on global supply chain.

The development of the SEAPORTS architecture involves different competencies (i.e. Modeling & Simulation, Risks analysis, etc.). Regarding Modeling & Simulation, SEAPORTS research teams has already taken part to the development of simulation infrastructures of marine terminals: Bruzzone and Longo (2008) developed a synthetic 3D environment based on High Level Architecture for low cost operators training in marine ports. Similarly Longo (2007) and Bruzzone et al. (2007) propose marine ports simulated environment for training students, engineers and operators. The SEAPORTS research team also acquired knowledge in developing supply chain risks analysis (Longo and Massei, 2008; Longo and Mirabelli 2008; De Sensi et al., 2008).

In light of the literature investigation on marine terminals, the enhancement of protection capability and security can turn out to be the choke point of efficiency in this very complex logistic system. Among the difficulties, it appears that the design and management of security procedures has been quite experience-based and did not receive a great deal of attention. Operational best practices for security vary from marine port to marine port according to techni-

cal and organizational aspects that can be driven by different factors (types of risks, alert level, available sensors technologies, etc.).

As stated by the SEAPORTS project objectives, there is a great need to improve the current security procedures eventually identifying security gaps to be filled by re-design security procedures/infrastructures and evaluating the effects of security advances on the global supply chain.

3. THE SEAPORTS ARCHITECTURE

The main objective of SEAPORTS is to enhance protection capability and security against threats in marine ports. The primary focus of SEAPORTS is to support:

- review, test and certification of ports security procedures;
- security gaps identification;
- design and re-engineering of security procedures/infrastructures;
- evaluation of effects of security advances in marine ports on the global supply chain.

The SEAPORTS architecture is based on System of systems (SoS) type of approach and it is focused on three key architectural elements:

1. The Modelling & Simulation Infrastructure (MSI) of the hosting port, intended as virtual simulation environment capable of recreating all the marine port operations.
2. The Key Risks Modelling Platform (KRMP), intended as generator of risk scenarios in terms of virtual threats and risks;
3. Sensor Technology Management Platform (STMP), intended as integration tool including existing and future virtual sensors technologies that can provide situational awareness and protection capability within marine ports.

The integration of the three main components forms the SEAPORTS architecture as shown in figure 1.

The SEAPORTS main idea is to develop, on the basis of a real marine port scenario, a synthetic environment (intended as combination of more parts) that recreate the whole complexity of a marine port in terms of: (i) main operations; (ii) risks, threats and external actions affecting the marine port environment; (iii) ways of detecting them.

In particular the Modeling & Simulation Infrastructure (MSI) recreates in a virtual simulation environment the marine port operations (i.e. vessels arrival and departure process, berth and yard management, yard equipment allocation, etc.).

The Key Risks Management Platform (KRMP), on the basis of risks analysis, generates virtual risk scenarios (virtual risks and threats) that affect the port operations (gen-

erated by the MSI). The Sensors Technology Management Platform (STMP) recreates a virtual platform of sensors technologies that combined with the security procedures should detect risks and threats within the marine port. By recreating the whole complexity of a marine port the SEAPORTS architecture can be used for supporting (in a synthetic environment) the review, test and certification of the security procedures currently used, for security gaps identification, for design and re-engineering of security procedures/infrastructures and for the evaluation of effects of security advances ports on the global supply chain.

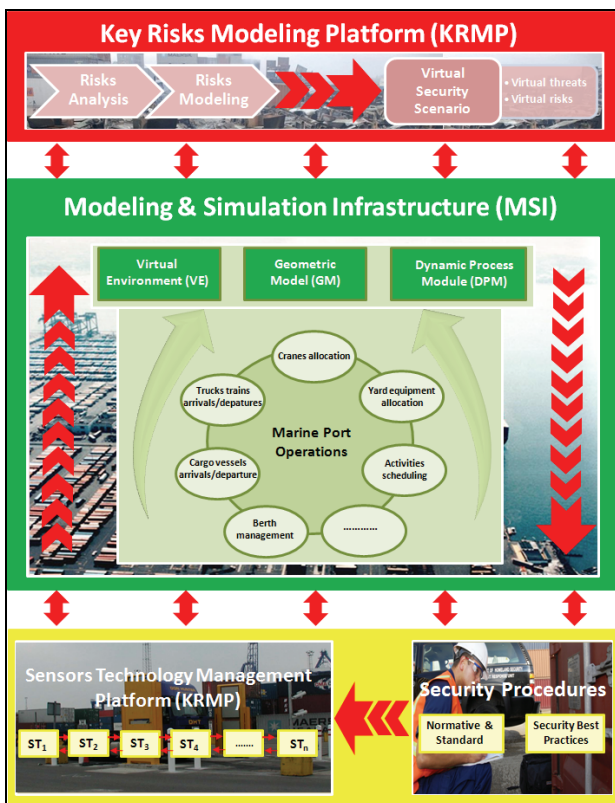


Figure 1: SEAPORTS architecture and main components integration

3.1 The Modeling & Simulation Infrastructure (MSI)

The MSI will recreate the evolution over the time of the port operations by using three integrated and cooperating modules: the *Virtual Environment (VE)*, the *Geometric Model (GM)* and the *Dynamic Processes Module (DPM)*. In effect if we look to a real marine port, we will note three integrated and cooperating components: a hosting environment (the real environment), the port lay-out (made up of berths, yard area, cranes, handling equipment, etc.), the dynamic interactions that evolve in the hosting environment and modify the port configuration as the time goes by (interactions generated by workers, containers movements, vessels arrival, etc.). In order to recreate the marine port, the SEAPORTS architecture integrates the three above-

mentioned components into the MSI. The translation into computerized models of the three cooperating components is as follows: the hosting environment is a *Virtual Environment (VE)*, the port lay-out is a three-dimensional *Geometric Model (GM)* and the interactions that evolve in the hosting environment and modify the port configuration are dynamic processes implemented within the *Dynamic Processes Module (DPM)*. Note that the DPM interacts with the GM and, in turn, the GM is contained into the VE. The integration of the DPM, the GM and the VE creates the MSI as shown in figure 1.

The MSI can be implemented by using different software tools. Note that the very final choice about software tools will be determined after the development of the port conceptual model. However, the SEAPORTS research team has already investigated the following solution. The VE should host and interact with port GM. The interactions that take place into the VE recreate the marine port operations; as before mentioned such processes are implemented within the DPM and the DPM could be implemented in a general purpose language (i.e. C++). As a consequence the VE should provide an interface with the C++ environment. To this end the *Vega Prime* platform (developed by *Presagis*) could be used for implementing the VE. Vega Prime is one of the most widely used tools for the creation of real time 3D application based on visual simulation. The software gives quite good performances on low cost hardware platform and its architecture is completely based on C++. The complete integration of the GM within the VE could be obtained by using the modelling toolset *Creator* (by *Presagis*). The *Creator* allows modelling objects, entities and sites and can be easily interfaced with Vega Prime.

3.2 The Key Risks Modeling Platform (KRMP)

A real marine port environment is characterized by several vulnerability points – such as port environment extension, access not adequately monitored, no complete containers inspections, entrance allowed also to unauthorized persons and is also subjected to threats as thefts, smuggling, terrorism, vandalism, and so on.

If, from one side, the MSI recreates in a virtual environment the main port operations, we need to recreate (within the virtual environment) risks and threats that characterize a marine port. To this end, SEAPORTS propose the development of the *Key Risks Modelling Platform* that acts as a generator of virtual risks and threats (virtual risk scenarios generation). The outline of the virtual risk scenarios have been defined by the SEAPORTS research team. It is worth saying that these virtual risk scenarios are by their very nature approximate and are not intended to limit the objectives of SEAPORTS and its ability to deal with a wide range of threats (note that the final virtual risk scenarios depends on the risks analysis in the port reference scenario and additional scenarios will be added during

the project development). Two virtual risk scenarios will be considered:

- the inclusion of threats within containers. The scenario will include a number of different virtual threats that can be easily sited in a container (radiological materials, weapons, etc.).
- the inclusion within the port area of threats delivered from the land side, such as threats entering the ports by trucks or trains or transported in other way by people (i.e. explosive, weapons, etc.).

In both cases the choice of the threats types will be based SEAPORTS research team available data and experience, their likelihood will be evaluated by the KRMP.

3.3 The Sensors Technology Management Platform (STMP)

A marine port environment is characterized by sensors technologies, security procedures and security plans devoted to reduce port vulnerabilities, increase protection capability and security. An approach devoted to support the review and redesign of security procedures and infrastructures (eventually identifying security gaps and evaluating the effects of security advances in marine ports facilities and global supply chain), must recreate sensors technologies devoted to detect risks and threats. To this end SEAPORTS propose to develop the *Sensors Technology Management Platform*. The STMP recreates virtual sensors technologies for detecting all the virtual risks and threats generated by the KRMP within the virtual marine port environment provided by the MSI.

Consider for instance the sensors technologies for containers inspection operations. Many container terminals are equipped with scanning machines such as the Vehicle and Cargo Inspection System (VACIS), based on gamma ray, radiation detection and Optical Character Recognition technologies. Thanks to the integration of such technologies the system is capable of identifying nuclear device or other radiological threats. In addition the VACIS is capable to detect people inside the container as well as highlight the container part that cannot be penetrated by the gamma ray. The presence of the virtual VACIS within the STMP could allow identifying (within the MSI) those containers that pose a risk in terms of nuclear or radiological threats.

4. SEAPORTS ARCHITECTURAL COMPONENTS INTEGRATION AND CASE STUDY DEVELOPMENT

The integration of the MSI, KRMP and SMTP recreate, in a synthetic environment, the whole marine port reference scenario in terms of virtual port operations (MSI), generation of virtual threats and risks within the marine port area (KRMP) and threats and risks detection by using virtual sensors technologies (STMP). The SEAPORTS architec-

tural components integration requires a research effort devoted to implement (within the Dynamic Processes Module of the MSI) all the security procedures. The security procedures, in turn, come out from an accurate analysis of normative and standards for marine port security and current best practices adopted by marine port's operators. Both normative and best practices analysis will give as result constraints and logical relationships (security procedures) for connecting the SEAPORTS main components. Note that marine ports usually implement different normative and adopt different best practices for port security. Consider for instance some marine ports have already implemented the Container Security Initiative (CSI) some others not, some ports have implemented the ISPS code, some others have implemented just the mandatory part. Similarly there are no standardized procedures for remedial actions in case of anomalies detection in the port area. Such "unclear" scenario is complicated by the presence of multiple actors each one with specific objectives. From the point of view of the container terminal management, security procedures should be kept at minimum (they negatively affect the marine terminal performance), from the point of view of government officers and Port Authorities security procedures should be completely integrated in the main port operations.

As first step we will integrate the SEAPORT architectural components on the basis of security procedures currently adopted in the port reference scenario, then we will evaluate alternative security procedures to be tested by using the SEAPORTS architecture.

The case study, developed on the basis of the port reference scenario, will aim at testing the SEAPORTS potentialities for supporting the enhancement of the protection capability and security of a marine port. In particular the case study will deal with the review, test and certification of the security procedures actually implemented within the port reference scenario, also identifying security gaps. Moreover SEAPORTS will be used for comparing the actual port scenario with alternative scenarios (design and re-engineering of security procedures/infrastructures). Finally SEAPORTS will be used for monitoring overall performance metrics devoted to evaluate the effects of security advances in the port reference scenario on the global supply chain.

5. CONCLUSIONS

The authors present the conceptual framework of an ongoing research project at the Modeling & Simulation Center – Laboratory of Enterprise Solutions (MSC-LES), at University of Calabria, Italy. In particular the paper proposes an advanced architecture called SEAPORTS to enhance protection capability and security against threats in marine ports (specifically container terminal). Note that it is not the scope of the SEAPORTS research program to develop a ready-to-use security platform to be implemented within a marine port area (i.e. a surveillance system or a security

system for marine port infrastructures). SEAPORTS is an advanced approach for investigating and analyzing the security problem within marine ports environment in order to find out security gaps and proceed with the redesign of the most critical security procedures and infrastructures. SEAPORTS also investigates the effects of port security procedures on global supply chain. To this end, the final result of the SEAPORTS research program will be a demonstrator (based on a real case study) that shows the potentialities of the proposed approach.

References

- Babul, M. J., 2004. No silver bullet: managing the ways and means of container security. United States Army Reserve USAWC strategy research project.
- Banks, J., 1998. Handbook of simulation, New York: Wiley Interscience.
- Bocca, E., Longo, F., Mirabelli, G., Viazzo, S., 2005. Developing Data Fusion systems devoted to security control in port facilities. In Proceedings of Winter Simulation Conference, pp. 445-449, Orlando, Florida, USA.
- Brown, R., 2003. Advanced technology for cargo security. The Third Meeting Inter-American Committee on Ports (CIP) and the Fourth Meeting of the Technical Advisory Group on Port Operations of the OAS.
- Bruzzone A.G., Longo F., Papoff E., 2005. Metrics for global logistics and transportation facility information assurance, security, and overall protection, Proceedings of 17th European Simulation Symposium, October 20th – 22nd, Marseille, France.
- Buzzzone A.G., Longo F., Massei M., Saetta S., 2006. The Vulnerability of Supply Chains As Key Factor in Supply Chain Management, Proceedings of Summer Computer Simulation Conference, pp. 181-186, July 30th – August 03rd, Calgary Canada.
- Bruzzone A. G., Longo F., 2008. VIP - Virtual Interactive Port. Proceedings of Summer Computer Simulation Conference, 15-18 June. Edinburgh (UK).
- Bruzzone A.G., Bocca E., Longo F., Massei M., 2007. *Logistics Node Design and Control over the Whole Life Cycle based on Web Based Simulation*, International Journal of Internet Manufacturing and Services, Vol. 1, Issue 1, pp 32-50.
- De Sensi G., Longo F., Mirabelli G., 2008. Inventory policies analysis under demand patterns and lead times constraints in a real supply chain, International Journal of Production Research, Vol. 46, Issue 24, pp 6997-7016.
- Dunn, R.H., 1987. The quest for software reliability, in Schulmeyer, G. G., McManus J. I., (Eds). Handbook of software quality assurance. New York: Van Nostrand Reynold, pp 342 – 384.
- Gambardella, L.M., Mastrolilli, M., Rizzoli, A.E., Zaffalon, M., 2001. An optimization methodology for intermodal terminal management. Journal of Intelligent Manufacturing, 12 (5-6), 521- 534.
- Kia, M., Shayan, E., Ghotb, F., 2002. Investigation of port capacity under a new approach by computer simulation. Computers & Industrial Engineering, 42 (2-4), 533- 540.
- Lewis, B., Erera, A., White, C., 2002. Efficient Container Security Operations at Transshipment Seaports. In Proceedings of the Inaugural Annual Intelligent Transportation Society (Singapore) Symposium.
- Longo, F., Bruzzone, A.G., 2005-a. Modeling & Simulation applied to Security Systems. In Proceedings of Summer Computer Simulation Conference, pp 183-188, Philadelphia, Pennsylvania, USA.
- Longo F., Mirabelli G., Viazzo S., 2005-b. Simulation and Design of Experiment for analyzing security issues in container terminals, Proceedings of the International Workshop on Modeling and Applied Simulation, pp 47-52, October 16th – 18th, Bergeggi (SV), Italy.
- Longo F., Mirabelli G., Bruzzone A.G., 2005-c. Container handling policy design by simulation framework, Proceedings of the International Conference Harbour, Maritime & Multimodal Logistics Modeling and Simulation, October 20th – 22nd, Marseille, France.
- Longo F., Mirabelli G., Briano E., Brandolini M., 2006. Container Terminal Scenarios Analysis And Awareness Trough Modeling & Simulation, Proceedings of 20th European Conference on Modeling and Simulation, May 28th – 31st, Bonn, Germany.
- Longo F., 2007. Economic Impact of a Terrorist Attack on a Railway System, in A.G. Bruzzone: *Security and Safety in Logistics*, Society for Computer Simulation International Eds, ISBN 1-56555-315-2.
- Longo F., Oren T., 2008. Supply Chain Vulnerability and Resilience: A state of the Art Overview. Proceedings of the European Modeling & Simulation Symposium, pp. 527-533, 17-19 September, Campora S. Giovanni (CS), Italy.
- Longo F., Massei M., 2008. Advanced Supply Chain Protection & Integrated Decision Support System. Proceedings of the Second Asia International Conference on Modeling & Simulation. Malesya, 13-15 May, pp 716 – 721.
- Longo F., 2007. Students training: integrated models for simulating a container terminal. Proceedings of 19th European Modeling & Simulation Symposium, October 04th – 06nd, 348-355, Bergeggi, Italy.
- Longo F., Mirabelli G., 2008. An Advanced Supply Chain Management Tool Based on Modeling & Simulation, Computer and Industrial Engineering, Vol 54/3 pp 570-588.
- Marine Log (2004)-a. MTSA and ISPS: final rules issued. Omaha: Simmons-Boardman Publishing Corporation.

- Marine Log, (2004)-b. Jitters as ISPS/MTSA deadline nears. Omaha: Simmons-Boardman Publishing Corporation.
- Montgomery, D. C., Runger, G. C., 2006. Applied Statistics and Probability for Engineers. Wiley Edition
- Moorthy, R., Teo, C.-P., 2006. Berth management in container terminal: the template design problem. *OR Spectrum*, 28 (4), 495-518.
- Oren T., Longo F., 2008. Emergence, Anticipation And Multisimulation: Bases For Conflict Simulation. Proceedings of the European Modeling & Simulation Symposium, pp. 546-555, 17-19 Settembre, Campora S. Giovanni (CS), Italy.
- Orphan, V. J., Muenchau, E., Gormley, J., Richardson, R., 2005. Advanced gamma-ray technology for scanning cargo containers, *Journal of applied Radiation and Isotopes*, 63, 723-732.
- Ottjes, J. A., Veeke, H., Duinkerken, M., Rijsenbrij, J., Lodewijks, G., 2006. Simulation of a multiterminal system for container handling. *OR Spectrum*, 28 (4), 447 – 468.
- Safety at Sea International (2003). What will be the cost of ISPS? As ports and shipping companies race to comply with an extremely tight deadline to meet the IMO security requirements. London: DMG World Media.
- Security Management (2003). IMO sets course for port security. Alexandria: American Society for Industrial Security.
- Shabayek, A.A., Yeung, W.W., 2002. A simulation model for the Kwai Chung container terminals in Hong Kong. *European Journal of Operations Research*, 140 (1), 1-11.
- Wenk, E., 2004. Container ports: security considerations, *Port Technology International*.

AUTHORS BIOGRAPHY

FRANCESCO LONGO took the degree in Mechanical Engineering from University of Calabria (2002) and the PhD in Industrial Engineering (2005). He is currently scientific responsible of the MSC-LES (Modeling & Simulation Center – Laboratory of Enterprise Solutions). He is also Associate Editors of the international journal *Simulation: Transaction of SCS*. He supports the organization of the most important conferences on M&S (SCSC, EMSS, MAS, etc.). His research interests include modeling & simulation of manufacturing systems and supply chain management, supply chain vulnerability, resilience and security, DOE, ANOVA.

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 include Modeling & Simulation for workstations effective design, work measurement and human reliability.

ALBERTO TREMORI is an Electronic Engineer, he took the degree from Genoa University; he acquired experience on the development of Simulators to support process re-engineering, logistics, transportation systems and warehouse automation. He is currently serving as President of MAST. He is member of the Simulation Team of DIPTTEM Genoa University and of Liophant Simulation.

TELESCOPE INSTRUMENT SIMULATOR MODELING FRAMEWORK FOR CONTROL SYSTEM DEVELOPMENT SUPPORT

Jose Maria Perez Menor^(a), Jose Carlos Lopez Ruiz^(a), Pablo Lopez Ramos^(a)

^(a)Instituto de Astrofisica de Canarias

^(a)jmperez@iac.es, ^(a)jlopez@ll.iac.es, ^(a)plopez@ll.iac.es

ABSTRACT

The development of astronomical instrumentation covers the development of the optic, mechanical parts, hardware devices and the control software system.

From the software point of view, we encounter a work environment where the control software must be developed while the hardware devices are not available. The hardware may take several years to be defined, designed, constructed and delivered. After the hardware arrival the control software must be developed in a short period of time. In most cases, hardware delays are tolerate. Real time hardware simulators can be used with the purpose of start software development at earlier project stages.

In this paper a modeling simulator framework for telescope instrumentation is presented. The main goal of the framework is to allow the beginning of the software development at the same time that hardware definition. So, more time is available for software development, quality and integration with telescope systems.

Keywords: Control System development, Instrument Simulator, Simulation and modeling framework, Real time simulators

1. INTRODUCTION

This work has been performed at the IAC (Instituto de Astrofísica de Canarias) Software Department, which is responsible of the control software for the instrumentation of several telescopes and satellites. Currently, several instruments for GTC (Gran Telescopio de Canarias) are being developed at the IAC.

The astronomical instrumentation must be integrated with telescope control systems (Filgueira and Rodriguez 1997). So, a software control layer that connects the telescope systems and the hardware must be developed.

The next instrument to be integrated with GTC will be EMIR (Espectrógrafo Multiobjeto Infrarrojo). EMIR is a wide-field, near-infrared, multi-object spectrograph, with image capabilities. It will allow observers to obtain many intermediate resolution spectra simultaneously, in the nIR bands Z, J, H, K. The instrument consists in

several custom-designed devices: Filters, Grisms, Translation Unit, Configurable masks, Window Lids, sensors, etc. All those elements are integrated inside a cryostat at 77 K°.

The devices are still under development, so we cannot use them to develop the software control system. The proposed simulation framework tries to solve this problem in the EMIR project and future developments.

Hardware device simulation is a fundamental task addressed when it is necessary to ensure the correct functionality of a control system, before actually plugging it into the real hardware.

Usually, the software must be developed while the hardware devices are not yet available to be used. This causes that the software control system must wait for the hardware. After the hardware arrival, all the pressure is put in the software development that has to start almost from scratch.

It is desirable to have a parallel development of hardware and control software, with the help of simulators. Additionally, the simulator should include some logic to emulate in a better way the real device behavior.

Another reason for the usage of simulators is the mean life time of a mechanism. The development of control software may wear out a mechanism due to the usage. It would be desirable to create the control system without running the real mechanism, so the development would not affect the mean life time.

The proposed framework provides the infrastructure and tools to describe mechanical, electronics mechanisms and sensors that must be controlled by the software, and to simulate and coordinate their behavior.

This earlier development also provides a way in designing and refining the mechanisms integration as their interactions with other mechanisms and the whole instrument .

The main objective of the framework is to speed up the development of instrument and mechanism simulators to support earlier software development. This also enhance software stability and quality as more time can be dedicated to those tasks.

Telescope and instrumentation simulators have evolved from simple structure programs to simulation frameworks.

A first evolution can be found in the object oriented paradigm, that facilitates modeling the structure and the behavior of the system by using “objects” that represent abstract entities to define design models (Douglas et al. 1993).

An evolution of objects can be found in the identification of design patterns in command and control software. This allows an evolution in the design of mechanism simulators. The inclusion of these design patterns in a modeling and simulation environment for specific domains plays a significant role in creating software design that can be simulated prior to have the real hardware (Ostroff, Dianic and Williams 2004; Noumaru, Mizumoto and Sasaki 1997).

The next evolution step is the creation of simulation frameworks that in combination with the usage of Domain Specific Languages (DSL) enhances the development of simulators (Ferayorni, Sarjoughian and Agutter 2007).

In the field of telescope and instrument simulators a differentiation must be made, between Optical Simulators that try to simulate the path and properties of the light through the lenses and mirrors (Davila, Bos, Breant and Edward 2008) and instrument behavior simulation, that focus in mechanisms simulation (Rodrigues, Correia, and Moreira 2008; Mora 2008)

Section 2 gives a global view of the framework developed. Section 3 presents the device simulator framework used to describe the mechanisms and their behavior. Section 4 presents the instrument simulator framework that integrates the device simulator and allows to define the global behavior. Finally, the conclusions and future works are presented.

2. SIMULATION AND MODELING FRAMEWORK

The model for a system is defined in terms of its structure and behavior.

The structure and organization of system components is modeled modularly and hierarchically.

The behavior of the system components can be modeled in continuous or discrete time according with our needs. The approach taken is a real time hybrid simulator where devices are simulated continuously and the instrument is simulated in a discrete way. The framework is focused on providing real time simulation to support the development of the final control system.

Each simulator provides an interface to the control software system using the same communication protocol as the final real hardware. The simulator must answer to the messages received in the same way as the real hardware, with the goal to change the simulator for the real hardware with much troubles.

2.1. Description

The framework provides tools to quickly develop devices simulators that are connected to the GCS

software control through simulated/real channels. But it might other control software system. It provides the mechanisms needed to isolate the simulators from the specific control software framework.

In the development of the proposed framework we try to encapsulate common functionality that do not depend on the business logic and provide the infrastructure in order that the simulators can benefit from them.

The framework provides: a Domain Specific Language to describe the simulators interfaces, properties and behavior. It also provides the mechanisms to simulate individual devices behavior, from the software control system point of view, and mechanisms that simulate the behavior of a full instrument (composed of several simulated devices).

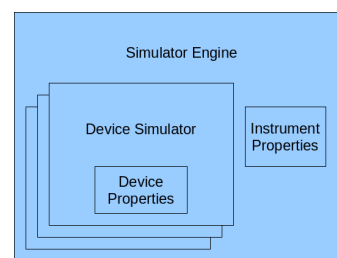


Figure 1: Basic Simulator Architecture

2.1.1. Device, Instrument and Channels

From the structural point of view the Instrument simulator is composed of Device simulators, properties and channels.

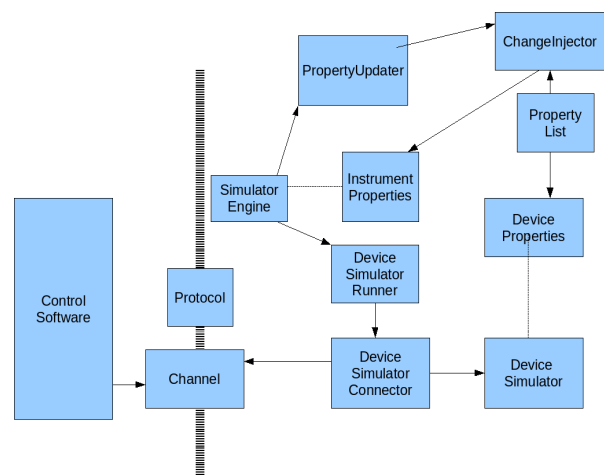


Figure 2: Simulator framework basic classes

The device and instrument simulator engine offers the possibility of loading those properties at initialization time. Those properties and their features are defined in configuration files: min value, default value, is constant, etc.

For more complex properties, as Motors, the configuration files allow to specify values as current, acceleration ramp, etc.

Device properties can be modified by the device simulator. For instance: the temperature property incremented when the motor is running. Or can be modified by the instrument simulator. For instance, the instrument is inside a cryostat with a cooling system, so the temperature decrease globally. The communication between the simulators and the clients is performed through channels.

3. DEVICE SIMULATORS

3.1. Simulator structure specification

From the structural point of view, a device simulator is defined by a Simulator class that implements the simulated behavior of the device, a connector class that receives the protocol messages and translate them in device operations and a channel class that simulate a real connection with the control software.

In order to facilitate that, the framework provides classes that already implement common behavior. In Table 1 the basic Simulator interface is presented. Some of those methods must be implemented obligatorily and other can be override according to the simulator needs.

Table 1: Provided Basic Simulator Interface

```

Simulator(const char* configFile);
virtual short start() = 0;
virtual short init() = 0;
virtual short stop() = 0;
virtual short abort() = 0;
virtual short shutdown() = 0;

virtual bool isMoving() = 0;
virtual short goToPosition() = 0;
virtual short goDefaultPosition() = 0;
virtual short stopMovement() = 0;

static void* callGoToDesiredPosition(void*
simulator);
static void* callGoDefaultPosition(void* simulator);
static void* callStopMovement(void* simulator);

virtual short calibrate() = 0;
virtual short powerOn() = 0;
virtual short powerOff() = 0;
virtual bool isPowerOn() = 0;

void setTriggeredMovement(bool enable);
bool isTriggeredMovement();

void setAutomaticPowerControl(bool value);
bool isEnabledAutomaticPowerControl();

Property* getProperty(char* name);

```

In the case that the simulator needs other methods or attributes a file definition must be used. From this file a C++ skeleton code for the simulator can be created. For instance, in Table 2 a definition of two additional methods is presented.

Also, the definition files allow to connect the simulator with the associated *Connector* in order to communicate with the desired channel. The connection between the *Simulator/Connector* and the used channel

is defined in the Instrument definition file (See section 4.1).

Table 2: Extended DTU Device Simulator interface

```

<simulator>
  <name>DTUSimulator</name>
  <method>
    <name>setDesiredPosition</name>
    <parameters>
      <parameter name="position">
        DTUPositionType
      </parameter>
    </parameters>
  </method>
  <method>
    <name>getDesiredPosition</name>
    <return>DTUPositionType</return>
  </method>
  <property>
    <name>position</name>
    <type>DTUPosition</type>
  </property>
</simulator>
<connector>
  <name>
    SerialPortToDTUSimulatorConnector
  </name>
  <property>
    <name>simulator</name>
    <type>DTUSimulator</type>
  </property>
</connector>

```

3.1.1. Properties Definition Files

Properties values can be modified without the need of further compilation. The properties are automatically read from a definition file and are created on the fly during the simulator initialization.

In that way the properties/variables of the simulated device can be modified according to the real hardware needs as it is being developed.

Those properties may represent a constant magnitude for configuration purpose, a variable magnitude for basic types (int, float, boolean, etc) or complex framework classes as the Motor class (See Table 3).

3.2. Behavior simulation

The device simulators can be executed in the instrument simulation environment or can run independently. Each of the device simulator is independent of the others.

The device simulators run in real time. This is because the main objective is the integration with the control system, so both must interact in real time. In that way, the simulator and the real hardware can be interchanged without timing problems.

The device simulator mainly waits for client requests or system events (for instance, temperature alarm). The *Connector* receives messages from the control system components through the specified channel and translates it in the specific Simulator Engine operations: Turn on power, init, move, etc.

Table 3: Properties configuration file

```

<properties>
  <property>
    <name>motor</name>
    <type>Motor</type>
    <value const="1">
      <topSpeed>7500.0</topSpeed>
      <accelRamp>25000.0</accelRamp>
      <decelRamp>50000.0</decelRamp>
      <boostCurrent>1.2</boostCurrent>
      <topSpeedCurrent>1.5</topSpeedCurrent>
      <stopCurrent>1.5</stopCurrent>
      <deltaStep>100</deltaStep>
    </value>
  </property>
  <property>
    <name>desiredStatus</name>
    <type>short</type>
    <value const="0" min="-1" max="1">0
    </value>
  </property>
  <property>
    <name>maxNumberOfSteps</name>
    <type>long</type>
    <value const="1">60000</value>
  </property>
  <property>
    <name>temperature</name>
    <type>float</type>
    <value const="0" min="70.0" max="150.0">
      77.0
    </value>
  </property>
</properties>

```

Those operations can be synchronous or asynchronous, as some devices may return control to the control software immediately after receiving a message. The framework provides support for both. For instance, the basic *Simulator* class provides two functions to start a device movement, one synchronous (*goToPosition*) and other asynchronous (*callGoToDesiredPosition*).

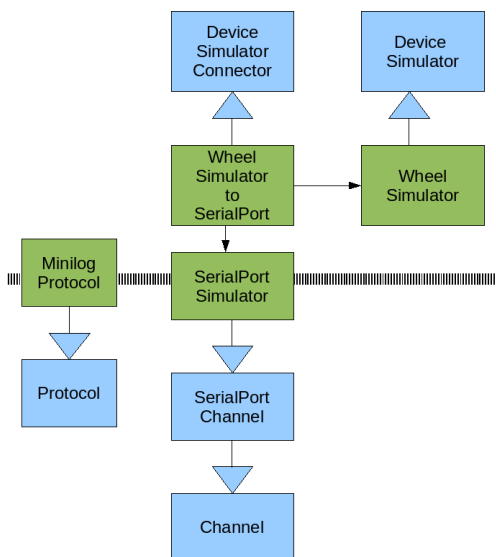


Figure 3: Simulator communication with clients

3.3. Communication with the clients

The clients (control software system) communicates with the simulator through channels.

The *Channel* class provides a basic implementation for the communication interface between the control software and the simulators. The framework provides support for several simulated connections (RS-232, CAN BUS, Ethernet, etc). The client and the simulators interchange message based on a device dependent protocol. Those protocols must be interpreted by the device simulator. Several base classes are provided (*Connector/Translator*) to facilitate the serialization, deserialization, interpretation and creation of response messages.

4. INSTRUMENT SIMULATOR

The instrument simulator provides an environment for executing all the device simulators and for the interaction between them. The Instrument simulator allows to define an environment with the goal of simulate the global behavior.

4.1. Simulator structure specification

The Instrument simulator is defined as an aggregation of devices, channels, properties and behavior functions (injectors). All those components are defined in a description file (See Table 4).

Table 4: Instrument simulator definition file

```

<properties>
  <property>
    <name>DTUSimulator</name>
    <type>Simulator</type>
    <channel>dev/com_dtu</channel>
    <triggeredMovement>1</triggeredMovement>
    <automaticPowerControl>
      1
    </automaticPowerControl>
  </property>
  <property>
    <name>CSUSimulator</name>
    <type>Simulator</type>
    <channel>dev/com_csu</channel>
    <triggeredMovement>0</triggeredMovement>
    <automaticPowerControl>
      1
    </automaticPowerControl>
  </property>
  <property>
    <name>temperature</name>
    <type>float</type>
    <value const="0" min="70.0" max="150.0">
      77.0
    </value>
  </property>
  <injector>
    <name>temperaturaUpdater</name>
    <type>TemperatureSimulatorInjector</type>
    <time>5</time>
    <property>
      <name>csu_temperature</name>
      <src>CSUSimulator</src>
    </property>
  </injector>
</properties>

```


It is possible to indicate what device simulators will be launch without recompile the simulator, only by changing configuration file.

As in device simulators, an instrument has properties that are defined in the configuration file. Those properties are used to simulate other important information apart from the devices. For instance, each device may have a temperature property and the instrument simulator may have a global temperature property.

The definition file can be used to run the instrument simulator directly or to generate code that can be extended.

The file is used to generate a C++ skeleton for the Simulator Engine, inheriting most of the functionality from the generic *SimulatorEngine* class.

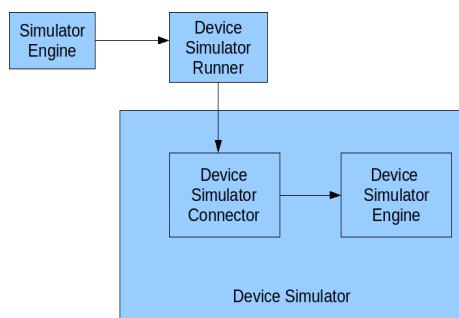


Figure 4: Run architecture for a device simulator

4.2. Behavior Simulator

In the dynamic part of the simulator two main functionalities can be identified: the simulator load and the simulated behavior. The first is performed through a class called *SimulatorEngine*. This class perform the following steps:

1. Read the definition file (properties and device simulators)
2. Create the indicated instrument properties and put them in a list
3. Create the indicated device simulators
4. Create the indicated channel-to-device_simulators connectors
5. Connect device simulators, connectors and channels
6. Start each device simulator through the *SimulatorRunner* class
7. Create change injectors
8. Add required device properties to each injector
9. Setup execution frequency for the injectors
10. Start to run the injectors through the *PropertyUpdater* class

Most steps are generic, with the only need to define the properties in the xml definition file and the implementation of the injectors. But as commented previously, C++ code can be generated for the instrument simulator and extend it in order to customize the loading process.

Once the instrument simulator is running, the properties are updated through Change Injectors. The injectors must implement a function that will be invoked periodically. The period for each injector is specified in the definition file.

In order to update the instrument properties, the Injector access instrument and device simulator properties. But an Injector cannot access the device simulators directly. Each injector has a Device/Properties list that is initialized during injector creation with references to the desired properties. The injector may also update the Device properties if necessary.

Most of the infrastructure needed by change injectors is provided by the *ChangeInjector* class. In order to implement the functionality to be executed periodically, a new class deriving from it must be created and a *handle_timeout* function must be override.

Table 5: Example of the temperature simulated behavior

```

Decrement = properties_["decrement"];

filterWheelTemperature =
    properties_["filterWheel_temperature"];
grismWheelTemperature =
    properties_["grismWheel_temperature"];
windowLid1Temperature =
    properties_["windowLid1_temperature"];
windowLid2Temperature =
    properties_["windowLid2_temperature"];
dtuTemperature =
    properties_["dtu_temperature"];
csuTemperature =
    properties_["csu_temperature"];
float devicesMean =
    (0.3 * filterWheelTemperature->value()
    +0.3*grismWheelTemperature->value()
    +0.05*windowLid1Temperature->value()
    +0.05*windowLid2Temperature->value()
    +0.1*dtuTemperature->value()
    +0.2*csuTemperature->value());
float updatedTemperature = ((temperature_->value()
    - decrement->value()) + devicesMean)/2;
temperature_->value(updatedTemperature);
// update device simulator temperatures
filterWheelTemperature->dec(decrement->value());
grismWheelTemperature->dec(decrement->value());
windowLid1Temperature->dec(decrement->value());
windowLid2Temperature->dec(decrement->value());
dtuTemperature->dec(decrement->value());
csuTemperature->dec(decrement->value());
  
```

In Table 5 a *TemperatureInjector* example is presented. This is the only code needed to simulate the cooling and heating part of the instrument system.

Also, each device simulator may update its values periodically or under some actions. For instance, heat is produced when a motor is working.

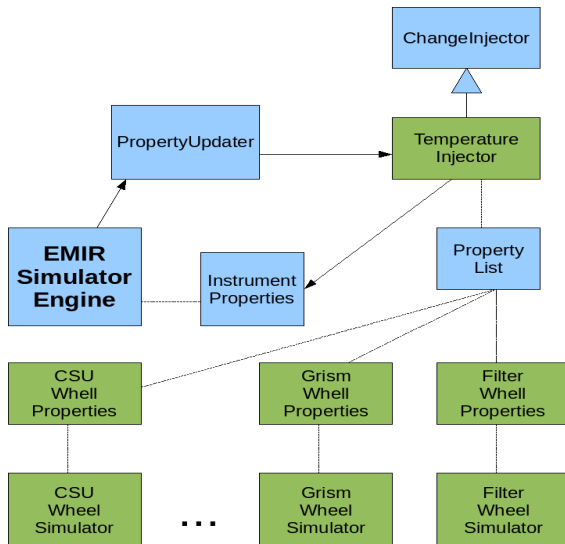


Figure 5: Detail of the EMIR Simulator architecture

5. CONCLUSIONS AND FUTURE WORKS

A framework for flexible and quick instrument simulator development has been presented.

Instrument and mechanism simulated devices can be configured through definition files to adjust to the real hardware when more information about it is available. For instance, the simulated device may pass without recompilation from a simulated filter wheel with six filters, each of the with their own features, to a new simulated wheel with eight filters with other values.

The framework presented is a valuable tool that allows the development of software control system at the same time as the hardware part of the instrument. In that way, more time can be dedicated to software development, testing and integration with the telescope control system.

Currently, the framework is focused on the GTC Control System, but it can be adapted and generalized to other control systems (telescope, satellites, etc) without much effort.

As future work, an Domain Specific Language for change injectors is going to be defined, in order to define simulator update operations without code.

Another issue for future work is automatic test generation (in python). With that feature apart of simulator testing, a test environment for the real hardware will be available.

REFERENCES

Davila, Bos, Brent and Edward, 2008. The Optical Telescope Element Simulator for the James Webb Space Telescope. *Proceedings of the SPIE, Volume 7010, pp 70103F-70103F-12 (2008)*.

Douglas et al., 1993 Object-Oriented Modeling and Simulation of Automated Control in Manufacturing. *IEEE International Conference on Robotics and Automation 1993: 83-88*. Atlanta, Georgia.

Ferayorni, Sarjoughian and Agutter, 2007. Domain driven simulation modeling for software design. *Proceedings of the 2007 summer computer simulation conference*. Pp. 297-304. San Diego, CA.

Filgueira and Rodriguez, 1997. GTC control system: and overview. *Proc. SPIE Vol. 3351, p. 2-12, Telescope Control Systems III*

Mora, 2008. Hardware device simulation framework in the ALMA Control subsystem. *ADASS XVIII*. Nov. 2-5, Québec.

Noumaru, Mizumoto and Sasaki, 1997. Test, debug and development environment of Subaru Observation Software System. *Proc. SPIE Vol. 3112, p. 52-59, Telescope Control Systems II*.

Ostroff, Dianic and Williams, 2004. Development and Operational Applications of a Real-time Range Data Simulator. *41st SpaceCongress*. April 27-30, 2004. July 24-28, Cape Canaveral, Florida.

Rodrigues, Correia, and Moreira, 2008. An Hybrid Design Solution For Spacecraft Simulators. *Proceedings of the Forum at the CAiSE'08 Conference*. June 18-20, 2008. Montpellier.

AUTHORS BIOGRAPHY

Dr. **Jose Maria Perez Menor** got the Computer Science degree from Madrid Polytechnic University in 2001 and the Ph.D. in Computer Science from University Carlos III of Madrid in 2006. Since 2008 he works for the Software Department of the Instituto de Astrofisica de Canarias as a Software Engineer. Formerly he was a Lecturer at the Computer Science Faculty in the University Carlos III of Madrid, mainly working in parallel file systems, high performance computin, Grid computing and network simulation.

José Carlos López Ruiz is a Software Engineer at the Instituto Astrofisica de Canarias where he works since 1998 building software for astronomy instruments. He got the Computer Science degree from Catalonia Polytechnic University in 1999 . Since 2005, he is also lecturer at the Computer Science Faculty in the University of La Laguna, mainly working on software engineering, advanced software engineering and databases.

Pablo López Ramos got the Computer Science degree from the Queens College of CUNY en 1986. Currently he is working in its Ph.D in Computer Science at the University of La Laguna, working in high performance computing. Since 1992 he has worked at the IAC as Systems Administrator and Software Engineer in the EMIR and OGS (Optical Ground Station) projects.

TESTS AND SIMULATION OF GAS BOILERS IN DOMESTIC HEATING PLANTS

Renzo Tosato^(a)

^(a)Department of Mechanical Engineering, University of Padova, Padova, Italy,

^(a)renzo.tosato@unipd.it

ABSTRACT

In E. U. and U.K. the SEDBUK label is the boiler equivalent of the energy label found on domestic appliances such as washing machines, fridges and dishwashers. SEDBUK stands for Seasonal Efficiency of a Domestic Boiler, and has lettered ratings from A to G. It's calculated from the results of standard laboratory tests together with other important factors such as boiler type and the kind of fuel used. The processes occurring in an apartment heating can be very different in comparison with a standard laboratory test and the SEDBUK of a boiler can be very far from actual seasonal efficiencies of the same boiler in different heating plants. The cyclical efficiency of traditional and condensing boilers can be mathematically 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. To simulate a boiler in a domestic heating plant and to calculate its seasonal efficiency a sinusoidal variation of loads during the season, from zero to a maximum value, can be assumed.

Keywords: Condensing Boiler, Seasonal Efficiency

1. INTRODUCTION

Energy-efficiency labels are informative labels affixed to manufactured boilers to describe the product's energy performance in the form of efficiency; these labels give consumers data necessary to make informed purchases.

Comparative labels allow consumers to compare performance among gas boilers using either discrete categories of performance or a continuous scale.

Examples of comparative labels are described in the Figure 1, and used in E.U. (Council Dir. 2005), U.S. (U.S. Department of Energy's Office, EERE, 200), Thailand, Australia, , Iran and Philippines.

In addition to giving information that allows consumers who care to select efficient models, labels also provide a common energy-efficiency benchmark that makes it easier for utility companies and government energy-conservation agencies to offer consumers incentives to buy energy-efficient products.

Energy-efficiency standards are procedures and regulations that prescribe the energy performance of gas boilers, sometimes prohibiting the sale of boilers that are less efficient than a minimum level. The term "standards" commonly encompasses the meaning: target

limits on minimum efficiency, that are formally established by a government, based on a specified test protocol.

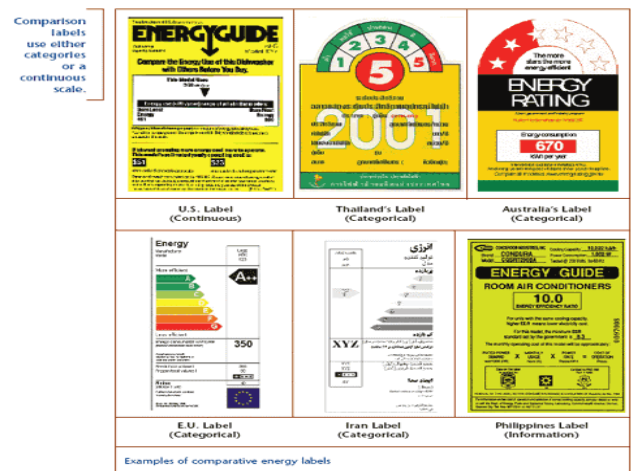


Figure 1: Examples of comparative labels

In E. U. and U.K. the SEDBUK label is the boiler equivalent of the energy label found on domestic appliances such as washing machines, fridges and dishwashers (Figure 2).

SEDBUK stands for Seasonal Efficiency of a Domestic Boiler, and has lettered ratings from A to G.

Band range	SEDBUK range	Band	SEDBUK
	90% and above		74% - 78%
	86% - 90%		70% - 74%
	82% - 86%		below 70%
	78% - 82%		

Figure 2: Letters and Seasonal Efficiency of Domestic Boilers

The rating represents the average annual efficiency achieved in typical domestic conditions, making reasonable assumptions about how it's used, the climate, controls, and other influences.

Each letter rating represents a range of percentage efficiencies with A (efficiency of more than 90%) being the best and G (efficiency of less than 70%) being the

worst. Conventional boilers are around 66-81% efficient, while condensing boilers are between 85% and 91% efficient.

It's calculated from the results of standard laboratory tests together with other important factors such as boiler type and the kind of fuel used.

This index, which was developed under the UK Government's Energy Efficiency Best Practice Programme with the help of boiler manufacturers, enables people to compare different models of boiler fairly.

New building regulations require that only gas boilers with Sedbuk A and B ratings can be installed unless it's too difficult or expensive, in which case an exception procedure must be completed and certified.

The processes occurring in an actual apartment heating can be very different in comparison with the standard laboratory test and the SEDBUK of a boiler can be very far from actual seasonal efficiencies of the same boiler in different years, heating plants, control system and regions.

Heating processes can be analyzed and simulated by computer programs. These programs examine various situations in order to evaluate heat losses during 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:

$$\text{Energy from heat emitters} = \text{GAS} \cdot H_i \cdot \eta_c \quad (1)$$

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 as the product of the three cyclical efficiencies which may be attributed to the three subsystems of the heating plant [2]:

$$\eta_c = \eta_{gc} \cdot \eta_{ec} \cdot \eta_{dc} \quad (2)$$

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.

In this paper the conventional and condensing boiler is taken into account, where $\eta_{dc} = \eta_{ec} = 1$. 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.

2. 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, flue gas, radiation and convection are major losses, which typically represent 10 to 20 % of

fuel input, (Natural Resources Canada's Office of Energ. Efficiency (OEE), 2006).

1) 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.

2) 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.

Considering a Model 1, the major energy losses associated with boilers fall into three categories: flue gas losses to the chimney P_{fl} during the period ON, radiation and convection losses of external surfaces P_{sl} during periods ON and OFF, ventilation losses to the chimney during period OFF presented in Figure 3.

For condensing boilers it is very difficult to evaluate these losses by experimental tests at the various loads, such as full, partial or null.

I prefer to consider for condensing and traditional boilers a different model (Model 2) presented in Figure 4, more close to the experimental conditions in a test rig (Rosa L., Tosato R., 1985),(Tosato R. 2008).

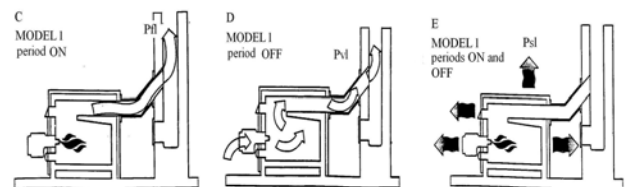


Figure 3: Model 1 of Energy losses during ON/OFF periods

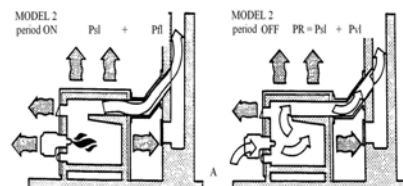


Figure 4: Model 2 of Energy losses during ON and OFF periods

A linear relationship between energy consumed and load is not considered as appropriate to calculate the energy consumption over a long period of time as during the winter .

$$P_{cons}/PN = A + B \cdot PU/PN \quad (3)$$

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.

The above 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.

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, alternatively to equation 3, the functions A and B (eq. 5 and 6) are not constant. Losses P_{fl} , P_{v1} and P_{s1} describe a linear equation on loads and can be assessed by experimental tests at full load (i.e. $P_{fl0}=0.2$ and $P_{s10}=0.1$) and at null load (i.e. $P_{v10}=0.3$). The results of this simulation (Model 1) are plotted in Figure 5.

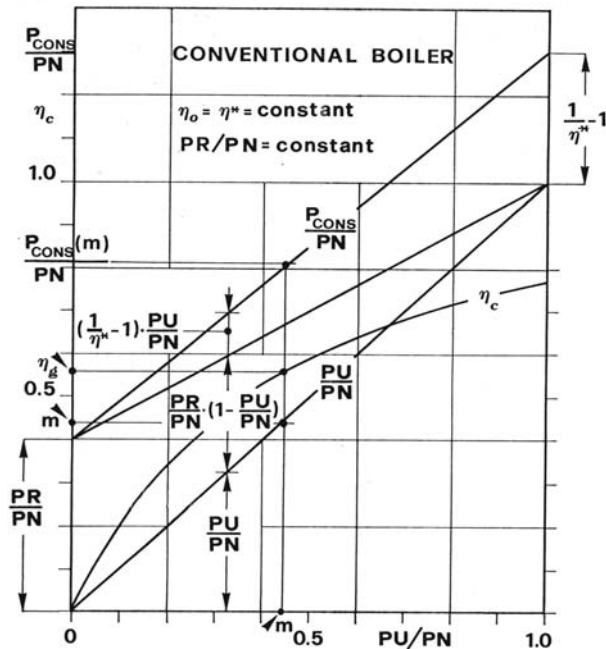


Figure 5: Acceptable model of a boiler at partial load when temperatures are more or less constant, using P_{fl} , P_{v1} and P_{s1} losses. (Model 1)

A second model (Model 2) of losses in a boiler is more adequate to the experimental tests. The major energy losses associated with boilers fall into two categories:

- flue gas losses to the chimney during the ON period, radiation and convection losses of external surfaces P_{s1} during the ON and OFF periods, ventilation losses to the chimney during the OFF period.

A model of a boiler can be defined by 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 (eqs. 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 Figure 6 for condensing boilers (Model 2) and in Figure 7 for traditional boilers (Model 2). The results of Figure 7 are the same as of Figure 5.

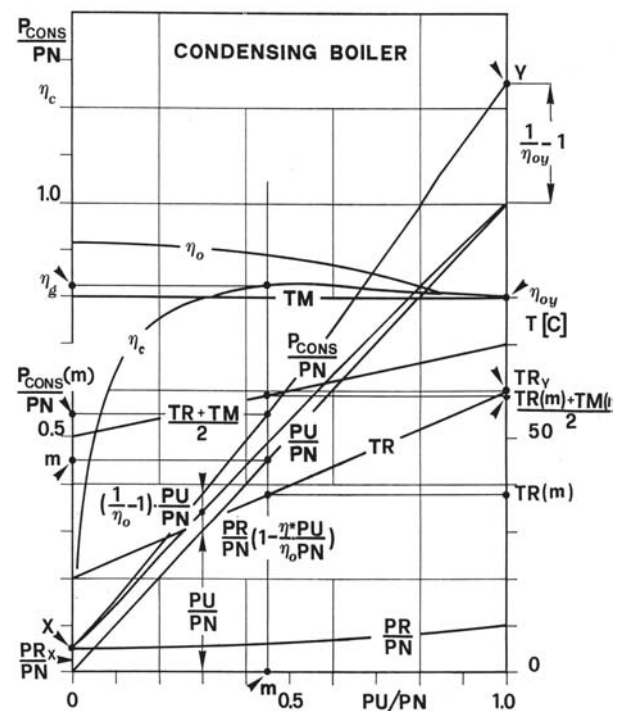


Figure 6: Model of boiler at partial load, using stand-by losses and efficiency η_0 at full load, considering their variation with water temperatures. Condensing boiler connected to plant/system by a 3-way mixing valve

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 Figure 8 (Model 2).

If the boiler temperature $(TR + TM)/2$ is taken as constant and the losses $P_{v1} = P_{v10} \cdot (1 - PU/PN)$ and $P_{fl} = P_{fl0} \cdot PU/PN$ are proportional to PU , according to Figure 5, then:

$$\frac{P_{CONS}}{PN} = \frac{1}{\eta_c} \cdot \frac{PU}{PN} = \frac{P_{fl}}{PN} + \frac{P_{v1}}{PN} + \frac{P_{s1}}{PN} + \frac{PU}{PN} = \frac{P_{fl0}}{PN} \cdot \frac{PU}{PN} + \frac{P_{v10}}{PN} \cdot (1 - \frac{PU}{PN}) + \frac{P_{s10}}{PN} + \frac{PU}{PN} =$$

$$\begin{aligned}
 &= (1/\eta_b - 1) \cdot (1 + P_{s10}/PN) \cdot PU/PN + P_{v10}/PN \cdot (1 - PU/PN) + \\
 &\quad + P_{s10}/PN + PU/PN = \\
 &= A + B \cdot PU/PN \quad (4)
 \end{aligned}$$

where η_b = combustion efficiency, and

$$A = P_{v10}/PN + P_{s10}/PN \quad (5)$$

$$\begin{aligned}
 B &= 1 + P_{f10}/PN - P_{v10}/PN = \\
 &= 1 + (1/\eta_b - 1) \cdot (1 + P_{s10}/PN) - P_{v10}/PN \quad (6)
 \end{aligned}$$

According to Figure 6, then it follows that:

$$\begin{aligned}
 P_{cons}/PN &= 1/\eta_c \cdot PU/PN = \\
 &= PR/PN \cdot (1 - \eta^*/\eta_o) \cdot PU/PN + (1/\eta_o - 1) \cdot PU/PN + \\
 &PU/PN = \\
 &= PU/(PN \cdot \eta) + PR/PN \cdot (1 - \eta^*/\eta_o) \cdot PU/PN \quad (7)
 \end{aligned}$$

where η^* is the nominal efficiency at full load, corresponding to the nominal return water temperature TR^* (usually 60 °C).

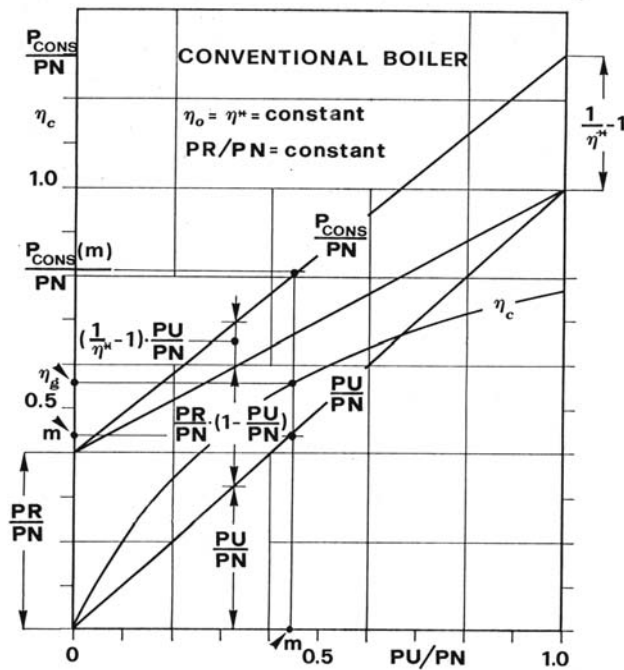


Figure 7: Acceptable model of a boiler at partial load; acceptable when temperatures are more or less constant, using stand-by losses PR and steady efficiency η_o

As represented in Figure 7, if $\eta_o = \eta^*$, η_o and PR are constant (as boilers at constant water temperature), the linear relationship, eqn. (4), for the cyclical efficiency, is:

$$P_{cons}/PN = 1/\eta_c \cdot PU/PN = PR/PN + (1/\eta^* - PR/PN) \cdot PU/PN = A + B \cdot PU/PN \quad (8)$$

where, according to eqns. (5) and (6):

$$A = PR/PN = P_{v10}/PN + P_{s10}/PN \quad (9a)$$

$$B = 1/\eta^* - PR/PN = 1 + P_{f10}/PN - P_{v10}/PN =$$

$$= 1/\eta_b \cdot (1 + P_{s10}/PN) - P_{s10}/PN - P_{v10}/PN \quad (9b)$$

$$1/\eta^* = 1/\eta_b \cdot (1 + P_{s10}/PN) \quad (10)$$

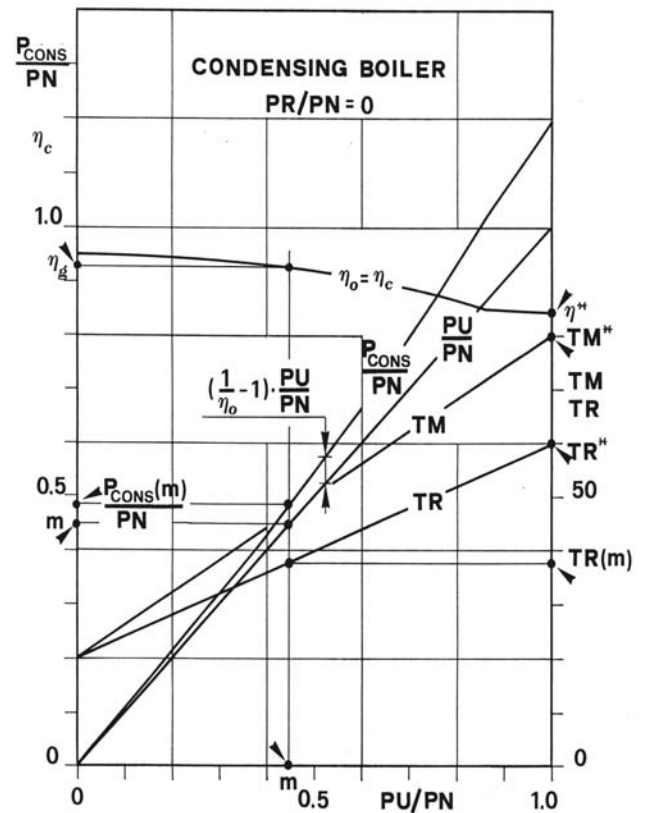


Figure 8: Model, using stand-by losses PR and steady efficiency η_o of a condensing boiler without stand-by losses, directly connected to a plant having $TR_d = TR^*$.

This result is not accurate for condensing boilers, because it presents a difference between η_o 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:

$$\eta_c = \eta_o \quad (11)$$

This efficiency (Figure 8) is not linear because η_o varies with the return water temperature.

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 elongated and characterized by low pressure drops. For such boilers the relationship applied was that of eqn. (7), which, obviously, is not linear and it depends on \square , 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 Figure 6, where, again, the relationship is not linear.

3. SEASONAL EFFICIENCY

The amount of gas consumption in an old constructed apartment, was collected during an eight-year period. This apartment measures 100 m² (AREA) and is heated by a 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 total annual consumption GASy = (828-2250) m³ and the profile of GAS consumption, beginning from September, are shown in Figure 9.

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 m³/m².

Figure 10 describes the nondimensional values of GAScons/GASy, which is variable only during the season and can be represented by a simple sinusoidal equation (12).

As a result, during a season, maximum loads LUm_{ax} of the boiler are required in a short period during the winter and LUm_{ax} values can be much lower than 100 %. In the other months LU can be assumed to vary from zero to LUm_{ax} with a cosine law (13).

$$\text{GAScons/GASy} = 50 \cdot (1 + \sin(\text{days} - \text{days}_0/2) \cdot 6,28/365/1,3)) \quad [\%] \quad (12)$$

$$\text{LU/LUm}_{\text{ax}} = 100 \cdot \cos((\text{days} - \text{day}_0/2) \cdot 6,28/365/1,3) \quad [\%] \quad (13)$$

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:

$$\eta_g = \eta_c(m) \quad (14)$$

In order to predict the seasonal efficiency of a boiler η_g an accurate result can be gained by using the following procedure:

(1) Define two experimental curves (not dependent upon the regulation system) representing the boiler efficiency at full load $\eta_0(\text{TR})$ function of TR and the losses $PR((\text{TR} + \text{TM})/2)$, by laboratory tests, as function of water mean temperatures $\text{TR} + \text{TM}/2$

(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) Apply N+1 loads from 0, $1/(N) \cdot \text{LU}_{\text{max}}$, $2/(N) \cdot \text{LU}_{\text{max}}$, ... to LU_{max} . As an example, given

$N=4$: (0, 0.25, 0.5, 0.75, and 1) * LU_{max} , calculate the corresponding values of cyclic efficiencies for the boiler;

(5) Assuming a sinusoidal variation of loads during the season from zero to LU_{max} calculate N intervals of days corresponding to the same loads: T1, T2, T3 and T4;

(6) Use the equation

$$\eta_g = \frac{\sum_{i=0}^{N-1} \eta_c \cdot \text{LU} \cdot T + \sum_{i=1}^N \eta_c \cdot \text{LU} \cdot T}{\sum_{i=0}^{N-1} \text{LU} \cdot T + \sum_{i=1}^N \text{LU} \cdot T} \quad (18)$$

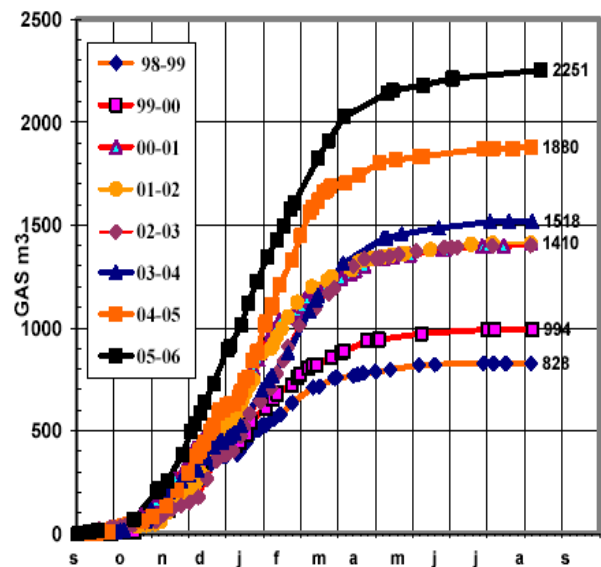


Figure 9: Profile of consumption of GAS

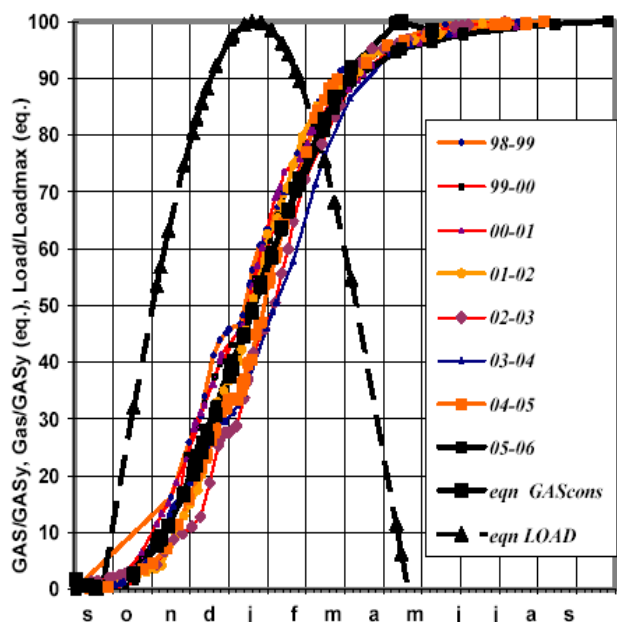


Figure 10: Profile of adimensional of GAS/GASy and LU/LUm_{ax}.

3. CONCLUSION

Sinusoidal loads from zero to maximum load (less than 100%) during a season seem to be acceptable in an apartment heating plants. The relationship between energy consumption and 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.

The energy labels SEDBUK (Seasonal Efficiency of a Domestic Boiler) used for boilers In E. U. and U.K. and equivalent of labels found on domestic appliances such as washing machines, fridges and dishwashers, can be very far from actual seasonal efficiencies of the same boiler in different heating plants. SEDBUK Is calculated from the results of standard laboratory tests. The processes occurring in an apartment heating can be very different in comparison with a standard laboratory test

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 with the usual regulation systems provided 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% than the efficiency of traditional boilers.

- The seasonal efficiencies of condensing boilers increase as the load decreases, while at the same condition the efficiencies of traditional boilers decrease.

- It is of considerable importance for both, the environmental protection and for energy saving to utilize condensing boilers: SO_x, NO_x, dust and soot, etc., which are the constituents of the flue gas; they can dissolve in the condensed water, and the pollutants emitted to the environment can be noticeably reduced.

4. NOMENCLATURE

AREA	surface of a 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,	[-]

ACKNOWLEDGMENTS

This work has been supported by Italian Ministero Istruzione Università Ricerca MIUR funds. Their contribution is gratefully acknowledged.

REFERENCES

- COUNCIL DIRECTIVE 92/42/EEC of 21 May 1992 on efficiency requirements for new hot-water boilers fired with liquid or gaseous fuels (version 11.8.2005).
- Natural Res. Canada's Office of En. Eff. (OEE), 2006, <http://oee.nrcan.gc.ca/industrial/equipment/boilers>
- 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
- Tosato R., MATHEMATICAL MODELS OF GAS FIRED BOILERS, 20th EMSS, 2008 European Modeling and Simulation Symposium (Simulation in Industry) September, 17-19, 2008, Campora San Giovanni, Amantea (CS) Italy
- U.S. Department of Energy's Office of Energy Efficiency and Renewable Energy (EERE). www.eere.energy.gov

AUTHORS BIOGRAPHY

Renzo Tosato: c/o the University of Padova: MSc in Mechanical Engineering, 1968; 1968-1980 Assistant Professor at the Dept. of Mechanical Engineering, Present position: since 1980, Associate Professor of "Fluid Machines" at the Dept. of Mechanical Engineering, University of Padova.

Research activity:

- experimental analysis and simulation of efficiency of gas boilers;
- experimental and numerical analysis of alternative compressors and unsteady gas flow.

A SEMANTIC APPROACH TO SIMULATION COMPONENT IDENTIFICATION AND DISCOVERY

Marianela García Lozano^(a), Farshad Moradi^(b), Eneko Ibarzabal^(c), Edward Tjörnhammar^(d)

^(a,b,d)FOI, Swedish Defence Research Agency, 164 90 Stockholm, Sweden

^(c)KTH Royal Institute of Technology, 100 44 Stockholm, Sweden

^(a)garcia@foi.se, ^(b)farshad@foi.se, ^(c)eneko@kth.se, ^(d)edward@foi.se

ABSTRACT

Development of simulations is often a costly process that consumes a lot of time and resources. An appealing approach to reduce the costs involved is to reuse and recombine existing predefined models and components through a composition process.

This process is a complex task that involves four main steps: identification, discovery, matching and composition. The focus of this work is to show how different ontologies combined with semantic querying enables more accurate identification and discovery of components in the process. We present a methodology for achieving this goal and clarification through a use case.

Our preliminary results indicate that our approach is feasible, and semantic techniques contribute to both finer descriptions and search results.

Keywords: Model composition, Semantic description, Ontology, BOM

1. INTRODUCTION

Modelling and Simulation (M&S) is a well known tool and method for e.g. exploring phenomenon that are too costly or impossible to explore in the real world. However, the development of simulations is many times an expensive process that consumes a lot of time and resources. An approach to reduce costs is to reuse existing models, mainly via recombination (Oses, Pidd and Brooks 2004).

To combine components into models a composition process of several steps is required. First, it is necessary to have an idea of what is to be simulated, i.e. to have a conceptual model of the simulation. From this, both the kinds of components and the requirements that are needed for the simulation may be identified. Then, from a set of components, viable candidates need to be located and selected. Finally, a matching of discovered components and comparison of different composition alternatives to the requirements needs to be done (Moradi, Nordvallér and Ayani 2006).

While the later steps of the process have been subject to research, the earlier steps are more or less untouched territory. In this paper we focus on these first steps i.e. the simulation description, and the identification and discovery of components.

The goal of this work has been to improve these steps by identifying and selecting components as accurately as possible i.e. only selecting relevant components and reducing the total number of composition alternatives to check. Hence, making the steps more precise and automated.

We propose a refined simulation composition process introducing semantic identification and discovery of components by *i*) coupling Simulation Reference Markup Language (SRML) documents with domain ontology information thus allowing a parser to extract more accurate information on the simulation components sought *ii*) presenting a Base Object Model (BOM) ontology which we together with a domain ontology use to describe simulation components *iii*) storing the descriptions in a semantic based distributed repository, and then retrieving viable candidates by *iv*) using SPARQL Protocol and Resource Description Framework Query Language (SPARQL - a recursive acronym) queries built from parsing the SRML documents. We have tested and evaluated our approach through a set of case studies and examples, which are presented in this paper.

1.1. Composability

One of the main challenges when composing simulation components is the issue of composability. Composability has been defined as the capability to select and assemble reusable simulation components in various combinations into simulation systems to meet user requirements (Petty 2004).

When addressing composability there are several issues to take into account. Composability checking of components needs to be performed on several aspects, syntactic, semantic and pragmatic.

- Syntactic composability focuses on the implementation aspects of each component, e.g. checking in and out parameters (Szabo and Teo 2007).
- Semantic composability addresses whether the combined computation of the components in the model is semantically valid (Petty 2004).
- Pragmatic composability addresses whether the model is meaningful with regard to the context in which it is to be used (Hofmann 2002).

As the definition suggests, syntactic composability is handled at the technical level to make sure that components have the right interfaces to interact with each other. Semantic composability however should be addressed at conceptual level. This requires precise definition and specification of components' syntax and semantics in order to capture the basic requirements for matching and synthesizing a semantically meaningful composition of those components. Here is a need for a common methodology for specification of simulation components consisting of meta-models describing the components at different levels. In order to enable automatic matching of meta-models they should be formalized and structured using ontologies. This is to avoid misunderstanding and to provide unambiguous definitions as a basis for reasoning about syntactic and semantic validity of compositions (Moradi et.al. 2009).

1.2. Relevant Techniques

The common denominator for all composition infrastructures is that they all besides the components themselves require a component repository, composition and query tools, and meta-data standards (Bartholet et.al. 2004). In this work we have utilized the following technologies, standards and tools.

The Simulation Reference Markup Language, see (SRML Web Site), is a markup language based on XML and a reference standard for representing simulations. It has enough expressiveness to model almost anything for the purpose of a simulation. SRML has been used in this work as a format for describing simulations scenarios.

The Document Object Model (DOM) is an API for XML and HTML documents. It is used to represent the document's logical structure and the document object's interaction possibilities. It is also language-independent.

The Base Object Model concept, see (BOM Web Site), has been identified and developed within High Level Architecture (HLA), see (HLA IEEE Standard), as basic components for rapid development of HLA object models. The goal with BOM is to enable reusability, interoperability and composability. It is based on the assumption that simulation and model parts may be extracted and reused as building blocks or components.

Ontologies are the corner stone of the Semantic Web initiative. Semantic Web is an attempt to simplify search and query of information on the web, and make the information understandable by computers, hence facilitating communication between them. Ontologies can be used to create a common understanding/context between components and describe, among others, entities, hierarchies, relations and attributes (Miller et.al. 2004, Chenine, Kabilan and Garcia Lozano 2006).

Ontologies are described using languages such as, Resource Description Framework (RDF) (RDF Web Site) and Web Ontology Language (OWL) (OWL Web Site). RDF is basically a directed graph data format used to represent information. With OWL there is a

better possibility of defining restrictions in the properties that relate different classes and data types.

SPARQL (Editors Prud'hommeaux, Seaborne 2008) is an RDF query language that enables queries consisting of triple patterns, their conjunctions and disjunctions. Query results can be sets or RDF graphs.

Besides the components themselves one of the main parts of a composition infrastructure is a repository facility which provides management and storage of, and access to those components. SDR is a Semantic based Distributed Repository (Garcia Lozano, Moradi and Ayani 2007) developed with the aim to enable sharing of resources with an overlay architecture. Resources may be anything that can be described e.g. simulation models, components, software, hardware, etc. Discovery of these resources is a main functionality and to make discovery as precise as possible semantic techniques are utilized.

2. APPROACH

In order to achieve our objectives, presented in the previous section, we develop an automated system for semantically retrieving simulation components. We assume that: *i*) each component is represented by a BOM model, stored in a distributed repository and *ii*) the simulation scenario is described by an SRML file.

The idea is that with the information specified in an SRML file describing the entities and interactions taking part in a simulation, and by semantically describing each BOM stored in the system, the process should be able to identify and discover relevant components automatically. Three aspects are of importance here; first, how the parsing of the SRML file is done and how the components are identified, second, how BOM components can be described, such that the information given in the descriptions corresponds to the information retrieved from the SRML file, and third, how the query used to discover the sought after BOMs should be formulated.

For the purpose of enriching BOM descriptions with semantic information we have defined a BOM ontology, which is presented in the following section.

2.1. BOM Ontology

The BOM ontology is implemented to enable semantic discovery of the simulation components. It provides a means for agreeing on the meaning of the different parts of a BOM description and their relationship. Thus enabling the mapping of features described in SRML to BOM "language". The ontology is specified in OWL and describes the structure of the BOM parts, the restrictions and relationships, specifying what is and what is not allowed when describing BOMs.

A BOM description consists of four major parts; Model Identification, Conceptual Model, Model Mapping and HLA Object Model.

The *Model Identification* part keeps track of meta data information about the BOM component and is used to document key meta data about the component i.e. information such as use history, unsuccessful inclusions

in simulations, description, creation date, information about how to contact the BOM's developers, application domain and purpose, etc.

The *Conceptual Model* contains information about the behavior of the component and how it actually works. It has four sub-parts, Pattern of Interplay, State Machine Model, Entity types and Event Types. Pattern of interplay contains actions, entity types and event types that take part during an interaction between two components. State Machine Model models how entity types move from one state to another via actions and conditions that must be fulfilled.

The third part is the *Model Mapping*, which describes the mapping between entity and event elements from BOMs to their counterparts in HLA's object model.

The fourth part is the *HLA Object Model* that provides the HLA OMT information like HLA object and interaction classes, their attributes and parameters, HLA data types.

The semantic information of a BOM, which is our main concern here, is contained in the two first parts. In the ontology we describe these parts together with existing relationships in OWL. The individual parts are described with classes named correspondingly. The attributes, relationships between classes and constraints have been modelled with data and object type properties, and restrictions. We have also added a central class called BOMContext which defines a BOM component. The BOMContext is related to the other classes; Identification Model, Entity Type, Event Type, Pattern of Interplay and State Machine with the hasBOMContext and isBOMContextOf object properties. For further details on the BOM Ontology, see (Ibarzabal 2008).

2.2. Methodology

The steps in our methodology are as follows:

1. We start by parsing the SRML document where we identify the simulation elements (an SRML file may contain multiple simulations). Having done that we build a DOM tree containing the document's elements, and store it in memory. A DOM tree contains a hierarchy of node objects with tags representing the object's names.
2. In this step we extract the information required to build the query from the DOM tree. The step begins by selecting and iterating through the simulation objects. In the XML domain an SRML Simulation description, Item and Property corresponds to Document, Element and Attribute. When processing each simulation we retrieve the ItemClass elements and their attributes such as Name and other values.
3. To build the SPARQL query we map the items' names to the Entity Type class defined in the BOM ontology. Item's attributes are then mapped as Entity Attributes.
4. The final step is to send the SPARQL query to the simulation component repository, in our case SDR, and ask for the components matching our query requirements.

In order to narrow down the possible component matches by restricting the hits to the ones from the relevant domain we also use a domain ontology in our methodology.

Since SRML does not include a domain specification, we add a parameter to the SRML document, which is a link to the domain ontology for that simulation. The same ontology is also used in the SPARQL query and when describing simulation components (BOMs).

The steps of our Identification and Discovery methodology can be seen in Figure 1.

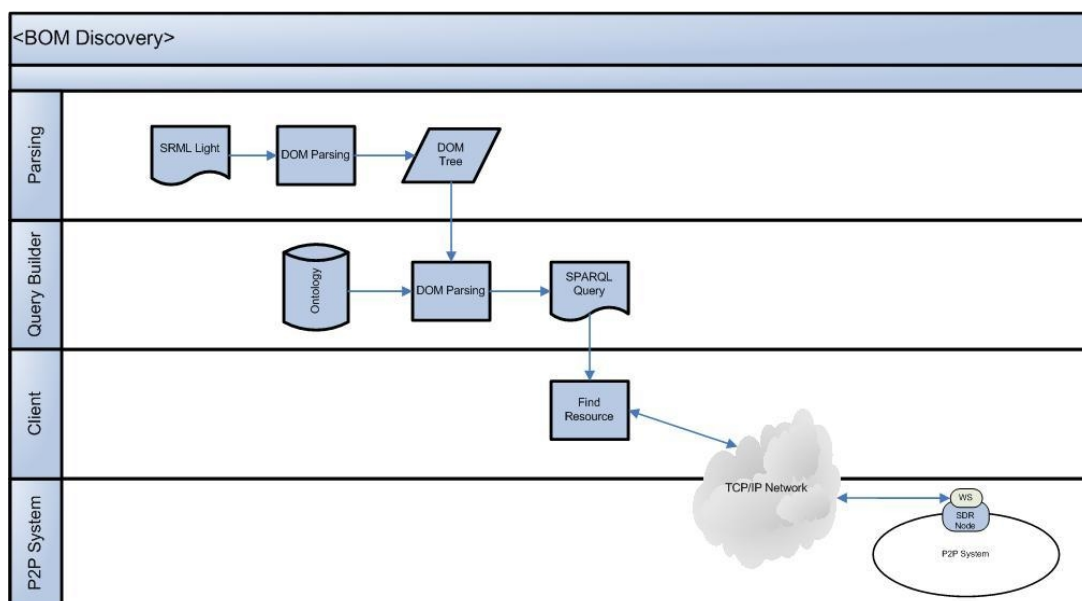


Figure 1: The process for identifying and discovering BOM models

3. USE CASES

To test our hypothesis, i.e. that it is possible to improve the simulation component identification and discovery with semantic techniques, we have built two different use cases. The first use case is based on a simulation requiring two models called “Tiger” and “Hawk”. Depending on the target domain they could represent (in the animal domain) a bird of prey and a cat, (in the military domain) a tank and helicopter, or (in the sports domain) the golf pro Tiger Woods and the skateboard pro Tony Hawk. A simple key word search without considering the domain of interest would result in a high recall with low precision. For example, searching for “Tiger” would result in many animal simulation models as well as tank and golf pro models. Taking the domain into consideration would yield lower recall and higher precision.

The second use case is based on the vehicle domain. In this use case we have different vehicle models like; bus, tractor, car and sports car. We also have models for the trips that can be made with the vehicles, and people interacting with them.

Listing 1: Small example of an SRML document

```

1.  <Simulation xmlns="urn:x-
2.  schema:SRML.xrdr"
3.  xmlns:pt="http://sdr.foi.se/ontology/
4.  PublicTransport.owl">
5.  <ItemClass name="pt:Person" />
6.  <ItemClass name="pt:Customer"
7.  SuperClasses="pt:Person">
8.  <pt:Customer pt:name="Jon"
9.  pt:age="29"
10. pt:travels="#BUS14">
11. <Script Type="text/javascript">
12. BroadcastEvent(all, "Greet",
13. "Good Morning");
14. SendEvent(Driver, "Pay");
15. </Script>
16. </pt:Customer>
17. </ItemClass>
18. <ItemClass name="pt:Driver"
19. SuperClasses="pt:Person">
20. <pt:Driver pt:name="Mike"
21. pt:drives="#BUS14">
22. </pt:Driver>
23. </ItemClass>
24. <ItemClass name="pt:Trip">
25. <pt:Trip pt:name="route66"
26. pt:hasSource="#Stockholm"
27. pt:hasDestination="#Uppsala">
28. </pt:Trip>
29. </ItemClass>
30. <ItemClass name="pt:BUS">
31. <pt:BUS pt:name="BUS14"
32. pt:hasDriver="#Mike"
33. pt:hasCustomer="#Jon"
34. pt:hasRoute="#route66">
35. <EventSink Name="Greet" />
36. </pt:BUS>
37. </ItemClass>
38. </Simulation>

```

To follow the methodology described in Section 2.2 we will begin by looking at a small example of an SRML document, see Listing 1. In this example we can see a simulation description containing different

people models i.e. a driver and a customer. There is also the vehicle type bus and trip model describing the route and destination of the bus. Note that we have added a name space called **pt**, which stands for the PublicTransport domain ontology, see line 3. Note also that in this example, due to space constraints, we use the pt name space in the attribute values even though that is not allowed according to the XML standard. In step 1 and 2 of the methodology the SRML file is parsed to extract the Items and their properties.

The public transport ontology used in the SRML file can be seen in Figure 2. It gives an overview of the classes and their relationships.

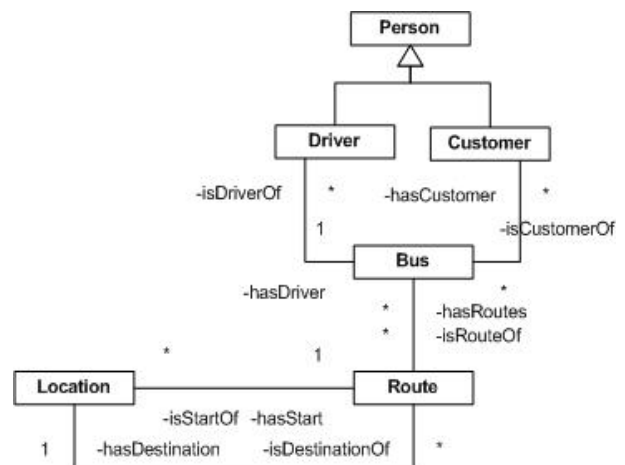


Figure 2: An ontology describing the public transport domain

Listing 2: The SPARQL Query Obtained from Parsing the SRML Document

```

1. PREFIX bom:
2. <http://sdr.foi.se/ontology/bom.owl#>
3. PREFIX pt:
4. <http://sdr.foi.se/ontology/
5. publicTransport.owl#>
6. SELECT ?x
7. WHERE {
8. {
9. ?x bom:isBOMContextOf ?x0 .
10. ?x0 bom:name "pt:Person" .
11. } UNION {
12. ?x bom:isBOMContextOf ?x1 .
13. ?x1 bom:name "pt:Customer".
14. ?x1 bom:isEntityTypeOf ?x10 .
15. ?x10 pt:name "Jon" .
16. ?x10 pt:age "29" .
17. ?x10 pt:travels "#BUS14" .
18. } UNION {
19. ?x bom:isBOMContextOf ?x2 .
20. ?x2 bom:name "Bus".
21. ?x2 bom:isEntityTypeOf ?x20 .
22. ?x20 pt:hasDriver "Mike" .
23. ?x20 pt:hasRoute "route66" .
24. }
25. }

```

After parsing the SRML file we move on to step 3 of the method where the resulting (slightly shortened) SPARQL query can be seen in Listing 2. The query will look for all BOMs that are of the right domain and contain the right kind of elements i.e. is either a Person,

Customer and/or BUS component. The original domain ontology is also supplied here as a PREFIX called pt. This will ensure that when we send the query to the semantic based component repository only BOM components from the correct domain will be returned.

In step 4 when we send the SPARQL query to SDR we get a list of the matching BOM components. An example of how a retrieved semantic BOM description could look can be seen in Listing 3. It describes a Bus model that also is a school bus. It also has different EntityAttributes like chassis and weight.

Listing 3: Example of a Semantic BOM Description

```

1. ...
2. <rdf:RDF ...
3.   xmlns:pt="http://sdr.foi.se/ontology/
   publicTransport.owl">
4.   xml:base="http://sdr.foi.se/ontology/
   bom.owl">
5. <BOMContext rdf:ID="busBOM1">
6.   <name rdf:datatype=
   "http://www.w3.org/2001/
   XMLSchema#string">
7.     Bus BOM1
8.   </name>
9.   <isBOMContextOf rdf:resource=
   "#SchoolBusA380"/>
10. </BOMContext>
11. <EntityType rdf:ID="SchoolBusA380">
12.   <name rdf:datatype=
   "http://www.w3.org/2001/
   XMLSchema#string">
13.     School Bus A380
14.   </name>
15.   <isEntityTypeOf rdf:resource=
   "#chassis"/>
16.   <isEntityTypeOf rdf:resource=
   "#weight"/>
17.   <hasBOMContext rdf:resource=
   "#busBOM1"/>
18.   <isSubClassOf>
19.     <BOMContextrdf:ID=
   "VehicleBOMContext"/>
20.   </isSubClassOf>
21. </EntityType>
22. <EntityAttribute rdf:ID="chassis">
23.   <name rdf:datatype=
   "http://www.w3.org/2001/XMLSchema#str
   ing">
24.     Chassis
25.   </name>
26.   <value rdf:datatype=
   "http://www.w3.org/2001/
   XMLSchema#string">
27.     RJ 45
28.   </value>
29.   <hasEntityType
   rdf:resource="#SchoolBusA380"/>
30. </EntityAttribute>
31. <EntityAttribute rdf:ID="weight">
32.   <name rdf:datatype=
   "http://www.w3.org/2001/
   XMLSchema#string">
33.     Weight
34.   </name>
35.   <value rdf:datatype=
   "http://www.w3.org/2001/
   XMLSchema#string">
36.     12 Tons
37.   </value>
38. </EntityAttribute>
39. ...

```

4. TEST AND EVALUATION

To make the tests we created different semantic BOM descriptions following the described use cases. These descriptions were then stored in SDR. Note: the BOM descriptions do not need to be stored in SDR but could be stored in any OWL compatible RDF triple store. In parallel we also made some matching SRML descriptions. Following the described methodology they were parsed and the resulting SPARQL queries were evaluated.

The results from our tests, even though preliminary, show that the domain specific semantic queries give an exhaustive, exact and relevant result set. There is however need for further tests and experiments. The next step will be to compare semantic queries with keyword based queries on a large data set in order to be able to draw better conclusions regarding benefits of using domain specific semantic searches.

However, one can already assume that less specific keyword based queries would have a lower precision and with a significantly higher recall, thus yielding more noise.

4.1. Evaluation

Based on our preliminary results semantic queries have higher accuracy rate and yield more relevant result set. However, in our experience they are harder to implement and if the set is a large one it is harder to eliminate hits. This is in part due to inefficiencies in SPARQL where we cannot create cascaded queries.

Key word based queries on the other hand are cheaper to evaluate but result in massive result sets with a lot of noise. They are also easier for a user to formulate.

Building ontologies is a cumbersome task which requires domain and application knowledge, and also modelling expertise. Semantic BOM building is not a trivial task either since it requires knowledge of both the BOM and domain ontology. The solution is to construct a tool which transparently aids in interleaving the BOM ontology with any other domain ontology.

5. SUMMARY AND CONCLUSIONS

To summarize our work we have refined the composition process by improving the simulation component identification and discovery. We have achieved this by using a semantic approach and automating the process. The method described has improved the relevance of the discovered simulation components. We have also automated the process by describing and storing the simulation components (BOMs) in a semantic distributed repository combined with an SRML parser. To enable semantic query evaluation we have built a BOM Ontology and introduced a new parameter to the SRML document allowing the inclusion of a domain ontology.

Based on this work our conclusions are that *i)* there exists a mapping between SRML and BOM descriptions, *ii)* it is possible to make a BOM ontology, *iii)* it is possible to automate simulation component

identification and discovery, and *iv*) it is possible to improve the relevance of simulation component discovery compared to not using semantic techniques.

REFERENCES

- Bartholet, R. G., Brogan, D. C., Reynolds, P. F. Jr. and Carnahan, J. C., 2004. In search of the philosopher's stone: Simulation composability versus componentbased software design. *In Proceedings of the 2004 Fall SIW*, Orlando, FL, BOM - Base Object Model. Web Site. <http://www.boms.info/> [Accessed June 2009]
- Chenine, M., Kabilan, M. V. and Garcia Lozano, M., 2006. A pattern for designing distributed heterogeneous ontologies for facilitating application interoperability. *EMOI-INTEROP*.
- Ibarzabal, E., 2008. *Enabling Semantic Discovery of Simulation Components*. Master's thesis, KTH, TRITA KTH/ICT/ECS 2008-40.
- Editors Prud'hommeaux, E. and Seaborne, A., 2008. SPARQL query language for RDF. Technical report W3C, January.
- HLA - High Level Architecture. IEEE Standard 1516.
- Hofmann, M., 2002. Introducing Pragmatics into VV&A. *In Proceeding of the European Simulation Interoperability Workshop*, London, UK,
- Garcia Lozano, M., Moradi, F. and Ayani, R., 2007. SDR: A Semantic based Distributed Repository for simulation models and resources. *In AMS'07: Proceedings of the First Asia International Conference on Modelling & Simulation*, pages 171-176, Washington, DC, USA, IEEE Computer Society.
- Miller, J. A., Baramidze, G. T., Sheth, A. P. and Fishwick, P.A., 2004. Investigating ontologies for simulation modeling. *In ANSS '04: Proceedings of the 37th annual symposium on Simulation*, page 55, Washington, DC, USA, IEEE Computer Society.
- Moradi, F., Nordvaller, P. and Ayani, R., 2006. Simulation model composition using BOMs. *In DSRT '06: Proceedings of the 10th IEEE international symposium on Distributed Simulation and Real-Time Applications*, pages 242-252, Washington, DC, USA, IEEE Computer Society.
- Moradi, F., Ayani, R., Mekarizadeh, S. and Tan, G. 2009. A rule-based semantic matching of Base Object Models, To be published in the *International Journal of Simulation and Process Modelling*.
- Oses, N., Pidd, M. and Brooks, R. J., 2004. Critical issues in the development of component-based discrete simulation. *Simulation Modelling Practice and Theory*, vol. 12, pp. 495-514.
- OWL - Web Ontology Language Overview. W3C Web site. <http://www.w3.org/TR/owl-features/> [Accessed June 2009]
- Petty, M. D., 2004. Simple composition suffices to assemble any composite model. *In Proceedings of*

the Spring Simulation Interoperability Workshop Orlando FL, April 18-23.

RDF - Resource Description Framework. W3C Web Site. <http://www.w3.org/RDF/>. [Accessed June 2009]

SRML - Simulation Reference Markup Language. W3C Web Site. <http://www.w3.org/TR/SRML/> [Accessed June 2009]

Szabo, C. and Teo, Y. M., 2007. On syntactic composability and model reuse. *In AMS '07: Proceedings of the First Asia International Conference on Modelling & Simulation*, pages 230-237, Washington, DC, USA, IEEE Computer Society.

AUTHORS BIOGRAPHY

Marianela Garcia Lozano is a Scientist at the Swedish Defence Research Agency (FOI). She holds a Master of Science in Computer Science from KTH, Royal Institute of Technology, and is now pursuing her PhD in the field of Semantic Resource Discovery in Distributed Systems. She has worked at FOI since 2001, with information and knowledge modelling, software development in distributed systems, Grids, Semantic Web, modelling and simulation, and Service Oriented Architectures.

Farshad Moradi is a senior scientist and programme manager in the area of modelling and simulation, at the Swedish Defence Research Agency. He has been working on modelling and simulation for the past 14 years and has led numerous projects in this area. He holds a Master of Science in Computer Science and Engineering from Chalmers University of Technology, and a PhD in Distributed Simulations from KTH, Royal Institute of Technology. His research interests are in the areas of Distributed Systems, Distributed and Web-based Modelling and Simulation, Embedded Simulations, Service-Oriented Architectures, Group behavioural modelling, and Logistics.

Eneko Ibarzabal holds a Master of Science in Engineering, Information and Communication Technology (with the major in Software Engineering and Distributed Systems) from KTH, Royal Institute of Technology, He did his master thesis at FOI with the topic of "Enabling Semantic Discovery of Simulation Components".

Edward Tjörnhammar is a research engineer at the Swedish Defence Research Agency. His research areas, and interests, are in distributed and parallel systems, information modelling and functional languages. He has also worked as a simulation technician at the Division of Air Combat Simulation (FLSC) and has experience in grid technology and network security. His studies were conducted at KTH, Royal Institute of Technology, in distributed and parallel systems.

ASYMPTOTIC STABILITY OF POSITIVE FRACTIONAL 2D LINEAR SYSTEMS WITH DELAYS

Tadeusz Kaczorek

Faculty of Electrical Engineering
Bialystok Technical University
Wiejska 45D, 15-351 Bialystok, Poland

kaczorek@isep.pw.edu.pl

ABSTRACT

The asymptotic stability of positive fractional 2D linear systems with delays is addressed. Necessary and sufficient conditions for the asymptotic stability of the positive fractional 2D systems with delays are established. It is shown that 1) the asymptotic stability of the positive fractional 2D systems does not depend on the number and values of the delays but only on the sum of system matrices, 2) the checking of the asymptotic stability of positive fractional 2D linear systems with delays can be reduced to testing the asymptotic stability of corresponding 1D positive linear systems.

Keywords: asymptotic stability, fractional, positive, 2D linear system, delay

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 200, Kaczorek 2002).

The most popular models of two-dimensional (2D) linear systems are the models introduced by Roesser (Roesser 1975), Fornasini-Marchesini (Fornasini-Marchesini 1976; 1978) and Kurek (Kurek 1985). These models have been extended for positive systems in (Kaczorek 1966; Valcher 1997; Kaczorek 2005; Kaczorek 2002). An overview of positive 2D systems theory has been given in the monographs (Farina and Rinaldi 2000; Kaczorek 2002). Reachability and minimum energy control of standard and positive 2D systems with one delay in states have been considered

in (Kaczorek 2005; Klamka 1991). The choice of the Lyapunov functions for positive 2D Roesser model has been investigated in (Kaczorek 2008c).

The notion of internally positive 2D system (model) with delays in states and in inputs has been introduced and necessary and sufficient conditions for the internal positivity, reachability, controllability, observability and the minimum energy control problem have been established in (Kaczorek 2006).

Stability of positive 1D and 2D linear systems has been considered in (Farina and Rinaldi 2002; Kaczorek 2008b; 2008c; Roesser 1975; Valcher 1997) and the robust stability in (Buslowicz 2008a). The positive fractional linear systems have been addressed in (Kaczorek 2008e; 2008f; 2007a; 2007b) and their stability has been investigated in (Buslowicz 2008b; Buslowicz and Kaczorek). LMI approaches to checking the stability of positive 2D systems have been proposed in (Kaczorek 2008d; Twardy 2007).

In this paper necessary and sufficient conditions for the asymptotic stability of the positive fractional 2D linear systems will be established and it will be shown that the asymptotic stability of the positive fractional 2D linear systems with delays can be reduced to testing the asymptotic stability of corresponding 1D positive linear systems.

2. POSITIVE FRACTIONAL 2D LINEAR SYSTEMS

Let $\mathfrak{R}^{n \times m}$ be the set of $n \times m$ real matrices. The set $n \times m$ matrices with nonnegative entries will be denoted by $\mathfrak{R}_+^{n \times m}$ and $\mathfrak{R}_+^n = \mathfrak{R}_+^{n \times 1}$. The set of nonnegative integers will be denoted by Z_+ and the $n \times n$ identity matrix will be denoted by I_n . A matrix $A = [a_{ij}] \in \mathfrak{R}^{n \times m}$ (a vector x) is called strictly positive and denoted by $A > 0$ ($x > 0$) if and only if $a_{ij} > 0$ for $i = 1, \dots, n$; $j = 1, \dots, m$.

Definition 1. The (α, β) orders fractional difference of an 2D function x_{ij} is defined by the formula

$$\Delta^{\alpha,\beta} x_{ij} = \sum_{k=0}^i \sum_{l=0}^j c_{\alpha\beta}(k,l) x_{i-k,j-l}, \quad (1a)$$

$n-1 < \alpha < n, \quad n-2 < \beta < n+1; \quad n \in N = \{1, 2, \dots\}$

where $\Delta^{\alpha,\beta} x_{ij} = \Delta_i^\alpha \Delta_j^\beta x_{ij}$ and

$$c_{\alpha,\beta}(k,l) = \begin{cases} 1 & \text{for } k=0 \text{ or/and } l=0 \\ (-1)^{k+l} \frac{\alpha(\alpha-1)\dots(\alpha-k+1)\beta(\beta-1)\dots(\beta-l+1)}{k!!l!!} & \\ \text{for } k+l > 0 \end{cases} \quad (1b)$$

The justification of Definition 1 is given in (Kaczorek 2008g).

Consider the (α, β) orders fractional 2D linear system with (q_1, q_2) -delays in state vector, described by the equations

$$\Delta^{\alpha,\beta} x_{i+1,j+1} = \sum_{k=0}^{q_1} \sum_{l=0}^{q_2} A_{kl}^0 x_{i-k,j-l} + A_{kl}^1 x_{i-k+1,j-l} + \quad (2a)$$

$$A_{kl}^2 x_{i-k,j-l+1} + B_0 u_{ij} + B_1 u_{i+1,j} + B_2 u_{i,j+1}$$

$$y_{ij} = Cx_{ij} + Du_{ij} \quad (2b)$$

where $x_{ij} \in \mathfrak{R}^n, u_{ij} \in \mathfrak{R}^m, y_{ij} \in \mathfrak{R}^p$ are the state, input and output vectors and $A_{kl}^t \in \mathfrak{R}^{n \times n}, k=0,1,\dots,q_1; l=0,1,\dots,q_2, B_t \in \mathfrak{R}^{n \times m}, t=0,1,2, C \in \mathfrak{R}^{p \times n}, D \in \mathfrak{R}^{p \times m}$.

Using Definition 1 we may write the equation (2a) in the form

$$x_{i+1,j+1} = \sum_{k=0}^{q_1} \sum_{l=0}^{q_2} \bar{A}_{kl}^0 x_{i-k,j-l} + \bar{A}_{kl}^1 x_{i-k+1,j-l} + \bar{A}_{kl}^2 x_{i-k,j-l+1} + \quad (3a)$$

$$+ \sum_{k,l \in D} c_{kl} x_{i-k+1,j-l+1} + B_0 u_{ij} + B_1 u_{i+1,j} + B_2 u_{i,j+1}$$

where

$$D = D_{i+1,j+1} \setminus D_{q_1 q_2}$$

and

$$D_{kl} = \{(i, j) \in Z_+ \times Z_+ : 0 \leq i \leq k, 0 \leq j \leq l\},$$

$$c_{kl} = -c_{\alpha\beta}(k, l)$$

$$\bar{A}_{kl}^0 = A_{kl}^0 + I_n c_{k+1,l+1} \quad \text{for } k=0,1,\dots,q_1; l=0,1,\dots,q_2$$

$$\bar{A}_{kl}^1 = \begin{cases} A_{kl}^1 + I_n c_{k,l+1} & \text{for } k=0; l=0,1,\dots,q_2 \\ A_{kl}^1 & \text{for } k=1,\dots,q_1; l=0,1,\dots,q_2 \end{cases}$$

$$\bar{A}_{kl}^2 = \begin{cases} A_{kl}^2 + I_n c_{k+1,l} & \text{for } k=0,1,\dots,q_1; l=0 \\ A_{kl}^2 & \text{for } k=0,1,\dots,q_1; l=1,\dots,q_2 \end{cases} \quad (3b)$$

The boundary conditions for the equation (3) have the form

$$x_{ij} = x_{ij}^b \in \mathfrak{R}^n \quad \text{for } i=0,-1,\dots,-q_1; j=0,-1,\dots,-q_2 \quad (4)$$

where x_{ij}^b are given.

Definition 2. The system (2)(and also (3)) is called the (internally) positive fractional 2D system if and only if $x_{ij} \in \mathfrak{R}_+^n$ and $y_{ij} \in \mathfrak{R}_+^p, i, j \in Z_+$ for any nonnegative boundary conditions (4) and all input sequences $u_{ij} \in \mathfrak{R}_+^m, i, j \in Z_+$.

Lemma 1.

a) If $0 < \alpha < 1$ and $1 < \beta < 2$ then

$$c_{kl} > 0 \quad \text{for } k=1,2,\dots; l=2,3,\dots \quad (5a)$$

b) If $1 < \alpha < 2$ and $0 < \beta < 1$ then

$$c_{kl} > 0 \quad \text{for } k=2,3,\dots; l=1,2,\dots \quad (5b)$$

Proof is given in (Kaczorek 2008g).

Theorem 1. The fractional 2D system (2) for $0 < \alpha < 1$ and $1 < \beta < 2$ (or $1 < \alpha < 2$ and $0 < \beta < 1$) is positive if and only if

$$\bar{A}_{kl}^t \in \mathfrak{R}_+^{n \times n}, B_k \in \mathfrak{R}_+^{n \times m},$$

$$k=0,1,\dots,q_1, l=0,1,\dots,q_2, t=0,1,2,\dots \quad (6)$$

$$C \in \mathfrak{R}_+^{p \times n}, D \in \mathfrak{R}_+^{p \times m}$$

The proof is similar to the proof for positive fractional linear systems without delays given in (Kaczorek 2009).

3. ASYMPTOTIC STABILITY

Definition 3. The positive fractional 2D linear system (2) is called asymptotically stable if for any nonnegative bounded boundary conditions (4)

$$\lim_{i,j \rightarrow \infty} x_{ij} = 0$$

for $x_{ij}^b \in \mathfrak{R}_+^n$ $i = 0, -1, \dots, -q_1$; $j = 0, -1, \dots, -q_2$ (7)

In the proof of the main result of this section the following lemma and theorem will be used.

Lemma 2. If $0 < \alpha < 1$ and $1 < \beta < 2$ (or $1 < \alpha < 2$ and $0 < \beta < 1$) then

$$\sum_{k=0}^{\infty} \sum_{l=0}^{\infty} c_{\alpha\beta}(k, l) = 0 \quad (8)$$

Proof. It is well-known (Buslowicz and Kaczorek) that

$$\sum_{i=0}^{\infty} (-1)^i \binom{\alpha}{i} = \sum_{i=0}^{\infty} (-1)^i \frac{\alpha(\alpha-1)\dots(\alpha-i+1)}{i!} = 0, \quad \alpha > 0 \quad (9)$$

Using (1b) and (9) we obtain

$$\begin{aligned} & \sum_{k=0}^{\infty} \sum_{l=0}^{\infty} c_{\alpha\beta}(k, l) = \\ & = \sum_{k=0}^{\infty} \sum_{l=0}^{\infty} (-1)^{k+l} \frac{\alpha(\alpha-1)\dots(\alpha-k+1)\beta(\beta-1)\dots(\beta-l+1)}{k!l!} = \\ & = \left(\sum_{k=0}^{\infty} (-1)^k \frac{\alpha(\alpha-1)\dots(\alpha-k+1)}{k!} \right) \times \\ & \times \left(\sum_{l=0}^{\infty} (-1)^l \frac{\beta(\beta-1)\dots(\beta-l+1)}{l!} \right) = 0 \end{aligned} \quad (10)$$

Theorem 2. The positive 2D general model with q delays

$$x_{i+1, j+1} = \sum_{k=0}^p \sum_{l=0}^q (A_{kl}^0 x_{i-k, j-l} + A_{kl}^1 x_{i-k+1, j-l} + A_{kl}^2 x_{i-k, j-l+1})$$

for $i, j \in Z_+$ (11)

where $x_{ij} \in R_+^n$ is the state vector and $A_{kl}^t \in R_+^{n \times n}$, $k=0, 1, \dots, p$, $l=0, 1, \dots, q$, $t=0, 1, 2$ is asymptotically stable if and only if the positive 1D system

$$x_{i+1} = \left(\sum_{k=0}^p \sum_{l=0}^q (A_{kl}^0 + A_{kl}^1 + A_{kl}^2) \right) x_i, \quad x_i \in R_+^n, \quad i \in Z_+ \quad (12)$$

is asymptotically stable.

Proof is given in (Kaczorek 2009; Kaczorek 2008b).

Theorem 3. The positive fractional 2D system (2) is asymptotically stable if and only if the positive 1D system

$$x_{i+1} = (\hat{A} + I_n) x_i, \quad \hat{A} = \sum_{k=0}^{q_1} \sum_{l=0}^{q_2} A_{kl}^0 + A_{kl}^1 + A_{kl}^2,$$

$$x_i \in \mathfrak{R}_+^n, \quad i \in Z_+ \quad (13)$$

is asymptotically stable.

Proof. From (3) for $B_0 = B_1 = B_2 = 0$ and (1) we have

$$x_{i+1, j+1} = \sum_{k=0}^{q_1} \sum_{l=0}^{q_2} A_{kl}^0 x_{i-k, j-l} + A_{kl}^1 x_{i-k+1, j-l} + A_{kl}^2 x_{i-k+1, j-l+1} + \sum_{k=0}^{i+1} \sum_{l=0}^{j+1} c_{kl} x_{i-k+1, j-l+1} \quad (14)$$

By Theorem 2 the positive 2D system with delays is asymptotically stable if and only if the positive 1D system

$$x_{i+1} = \left(\hat{A} + \sum_{k=0}^{\infty} \sum_{l=0}^{\infty} c_{kl} I_n \right) x_i, \quad x_i \in R_+^n, \quad i \in Z_+ \quad (15)$$

is asymptotically stable.

From (1b) we have $c_{00} = -c_{\alpha\beta}(0, 0) = -1$ and from (20) we obtain

$$\sum_{k=0}^{\infty} \sum_{l=0}^{\infty} c_{kl} I_n = I_n \quad (16)$$

Substitution of (16) into (15) yields (13). \square

Applying to the positive 1D system the well-known theorem (Kaczorek 2008b) we obtain the following theorem.

Theorem 4. The positive fractional 2D system (2) is asymptotically stable

if and only if one of the following equivalent conditions holds:

- 1) Eigenvalues z_1, \dots, z_n of the matrix $\hat{A} + I_n$ have moduli less than 1,
- 2) All coefficients of the characteristic polynomial of the matrix \hat{A} are positive,
- 3) All leading principal minors of the matrix $-\hat{A}$ are positive,

Theorem 5. The positive fractional 2D system (2) is unstable if at least one diagonal entry of the matrix \hat{A} is positive.

Proof. If at least one diagonal entry of the matrix \hat{A} is positive then at least one diagonal entry of the matrix $\hat{A} + I_n$ is greater than 1 and it is well-known (Buslowicz and Kaczorek; Galkowski 2001; Kaczorek 2006) that the system (25) is unstable. \square

Example 1. Using Theorem 4 check the asymptotic stability of the positive fractional 2D system (2) for $\alpha = 0.3$, $\beta = 1.2$ and $q_1 = q_2 = 1$ with the matrices

$$A_{00}^0 = \begin{bmatrix} 0.4 & 0.1 \\ 0 & 0.36 \end{bmatrix}, A_{00}^1 = \begin{bmatrix} -1 & 0.1 \\ 0 & -1.1 \end{bmatrix}, A_{00}^2 = \begin{bmatrix} -0.2 & 0.1 \\ 0 & -0.3 \end{bmatrix}$$

$$A_{01}^0 = \begin{bmatrix} -0.01 & 0.1 \\ 0 & -0.02 \end{bmatrix}, A_{01}^1 = \begin{bmatrix} 0.2 & 0.1 \\ 0 & 0.12 \end{bmatrix}, A_{01}^2 = \begin{bmatrix} 0 & 0.2 \\ 0 & 0.2 \end{bmatrix}$$

$$A_{10}^0 = \begin{bmatrix} 0.2 & 0 \\ 0 & 0.2 \end{bmatrix}, A_{10}^1 = \begin{bmatrix} 0.01 & 0.2 \\ 0 & 0.3 \end{bmatrix}, A_{10}^2 = \begin{bmatrix} -0.1 & 0.4 \\ 0 & -0.1 \end{bmatrix}$$

$$A_{11}^0 = \begin{bmatrix} -0.01 & 0.1 \\ 0 & -0.01 \end{bmatrix}, A_{11}^1 = \begin{bmatrix} 0.1 & 0.1 \\ 0 & 0.1 \end{bmatrix}, A_{11}^2 = \begin{bmatrix} 0.1 & 0 \\ 0 & 0 \end{bmatrix} \quad (17)$$

The fractional system is positive since the matrices

$$\bar{A}_{00}^0 = A_{00}^0 + I_n c_{11} = \begin{bmatrix} 0.04 & 0.1 \\ 0 & 0 \end{bmatrix},$$

$$\bar{A}_{40}^0 = A_{40}^0 + I_n c_{21} = \begin{bmatrix} 0.074 & 0 \\ 0 & 0.074 \end{bmatrix},$$

$$\bar{A}_{01}^0 = A_{01}^0 + I_n c_{12} = \begin{bmatrix} 0.026 & 0.1 \\ 0 & 0.016 \end{bmatrix},$$

$$\bar{A}_{11}^0 = A_{11}^0 + I_n c_{22} = \begin{bmatrix} 0.0026 & 0.1 \\ 0 & 0.0026 \end{bmatrix},$$

$$\bar{A}_{00}^1 = A_{00}^1 + I_n c_{01} = \begin{bmatrix} 0.2 & 0.1 \\ 0 & 0.1 \end{bmatrix},$$

$$\bar{A}_{01}^1 = A_{01}^1 + I_n c_{02} = \begin{bmatrix} 0.08 & 0.1 \\ 0 & 0 \end{bmatrix},$$

$$\bar{A}_{10}^1 = A_{10}^1 = \begin{bmatrix} 0.01 & 0.2 \\ 0 & 0.3 \end{bmatrix}, \bar{A}_{11}^1 = A_{11}^1 = \begin{bmatrix} 0.1 & 0.1 \\ 0 & 0.1 \end{bmatrix},$$

$$\bar{A}_{00}^2 = A_{00}^2 + I_n c_{10} = \begin{bmatrix} 0.1 & 0.1 \\ 0 & 0 \end{bmatrix}, \bar{A}_{01}^2 = A_{01}^2 = \begin{bmatrix} 0 & 0.2 \\ 0 & 0.2 \end{bmatrix},$$

$$\bar{A}_{10}^2 = A_{10}^2 + I_n c_{20} = \begin{bmatrix} 0.005 & 0.4 \\ 0 & 0.005 \end{bmatrix},$$

$$\bar{A}_{11}^2 = A_{11}^2 = \begin{bmatrix} 0.1 & 0 \\ 0 & 0 \end{bmatrix}$$

have nonnegative entries.

In this case

$$\hat{A} = \sum_{k=0}^1 \sum_{l=0}^1 (A_{kl}^0 + A_{kl}^1 + A_{kl}^2) = \begin{bmatrix} -0.31 & 1.5 \\ 0 & -0.25 \end{bmatrix}$$

The first condition of Theorem 4 is satisfied since the matrix

$$\hat{A} + I_n = \begin{bmatrix} 0.69 & 1.5 \\ 0 & 0.75 \end{bmatrix}$$

has the eigenvalues $z_1 = 0.69$, $z_2 = 0.75$ whose moduli are less than 1. The second condition of Theorem 4 is also satisfied since the characteristic polynomial

$$\det[I_n z - \hat{A}] = \begin{vmatrix} z + 0.31 & -1.5 \\ 0 & z + 0.25 \end{vmatrix} = z^2 + 0.56z + 0.0775$$

has positive coefficients.

The leading principle minors of the matrix

$$-\hat{A} = \begin{bmatrix} 0.31 & -1.5 \\ 0 & 0.25 \end{bmatrix}$$

i.e. $\Delta_1 = 0.31$, $\Delta_2 = 0.0775$. Therefore, all three conditions of Theorem 4 are satisfied and the positive fractional 2D system with (17) is asymptotically stable.

4. CONCLUDING REMARKS

Necessary and sufficient conditions for the asymptotic stability of the positive fractional 2D systems with delays have been established. Simple sufficient conditions for instability of the system have been also given. It has been shown that 1) the asymptotic stability of the positive fractional 2D systems does not depend on the number and values of the delays but only on the sum of system matrices, 2) checking the asymptotic stability of the positive fractional 2D linear systems with delays can be reduced to testing the stability of corresponding 1D positive linear systems.

Assuming in (1a) $B_1 = B_2 = 0$ we obtain the first Fornasini-Marchesini model and assuming $A_0 = 0$ and $B_0 = 0$ we obtain the second Fornasini-Marchesini model. Therefore, the considerations for the first and second F-M models are particular case of the presented

general 2D model. The considerations presented for the general model can be easily extended for the Roesser model. Extension of those considerations for the hybrid positive 2D linear systems and for 2D continuous-time linear systems are open problems.

ACKNOWLEDGMENTS

This work was supported by Ministry of Science and Higher Education in Poland under work No NN514 1939 33.

REFERENCES

- Bose N.K., 1985, *Multidimensional Systems Theory Progress, Directions and Open Problems*, D. Reidel Publishing Co.
- Buslowicz M., 2008a. Stability of linear continuous-time fractional order systems with delays of the retarded type. *Bull. Pol. Acad. Techn. Sci.* vol. 56, no. 4, 319-324.
- Buslowicz M., 2008b. Simple stability conditions for linear discrete-time systems with delays. *Bull. Pol. Acad. Techn. Sci.* vol. 56, no. 4, 325-328.
- Buslowicz M., Kaczorek T., Simple conditions for practical stability of positive fractional discrete-time linear systems. (in Press).
- Farina L., Rinaldi S., 2000. *Positive Linear Systems; Theory and Applications*. J. Wiley, New York.
- Fornasini E., Marchesini G., 1976. State-space realization theory of two-dimensional filters. *IEEE Trans. Autom. Contr.*, Vol. AC-21, 484-491.
- Fornasini E., Marchesini G., 1978. Double indexed dynamical systems, *Math. Sys. Theory*, 12, 59-72.
- Galkowski K., 2001. *State-space Realizations of Linear 2-D Systems with Extensions to the General nD (n>2) Case*. Springer, London.
- Kaczorek T., 2002. *Positive 1D and 2D Systems*, Springer-Verlag, London.
- Kaczorek T., 2008a. Asymptotic stability of positive 2D linear systems. *Proc of 13th Scientific Conf. Computer Applications in Electrical Eng.*, April 14-16, Poznan, Poland.
- Kaczorek T., 2008b. Asymptotic stability of positive 1D and 2D linear systems. *Recent Advances in Control and Automation*, Acad. Publ. House EXIT, 41-52.
- Kaczorek T., 2008c. Practical stability of positive fractional discrete-time linear systems, *Bull. Pol. Acad. Techn. Sci.* vol. 56, No. 4, 313-318.
- Kaczorek T., 2008d. LMI approach to stability of 2D positive systems with delays. *Multidimensional Systems and Signal Processing*, vol. 18, no. 3 (in Press)
- Kaczorek T., 1966. Reachability and controllability of non-negative 2D Roesser type models, *Bull. Acad. Pol. Sci. Ser. Sci. Techn.*, vol. 44, no 4, 405-410.
- Kaczorek T., 2005. Reachability and minimum energy control of positive 2D systems with delays, *Control and Cybernetics*, vol. 34, no 2, 411-423.
- Kaczorek T., 2008e. Reachability and controllability to zero tests for standard and positive fractional discrete-time systems, *Journal of Automation and System Engineering*. (in Press).
- Kaczorek T., 2007a. Reachability and controllability to zero of positive fractional discrete-time systems. *Machine Intelligence and Robotic Control*, vol. 6, no. 4.
- Kaczorek T., 2007b. Reachability and controllability to zero of cone fractional linear systems, *Archives of Control Sciences*, vol. 17, no. 3, 357-367.
- Kaczorek T., 2008f. Fractional positive continuous-time linear systems and their reachability. *Int. J. Appl. Math. Comput. Sci.*, vol. 18, no. 2 (in Press).
- Kaczorek T., 2006. Positive 2D systems with delays, *MMAR 2006, 12th IEEE IFAC International Conference on Methods in Automation and Robotics*, Poland.
- Kaczorek T., Practical stability of positive fractional 2D linear systems, *Bull. Acad. Pol. Sci. Ser. Sci. Techn.*, vol. 56, no. 4, 313-317.
- Kaczorek T., 2009b. Independence of the asymptotic stability of the 2D linear systems with delays of their delays. *Int. J. Appl. Math. Comput. Sci.*, vol. 19, (in Press).
- Kaczorek T., 2008g. Positive different orders fractional 2D linear systems, *Acta Mechanica et Automatica*, vol.2, no. 2, 51-58.
- Klamka J., 1991. *Controllability of Dynamical Systems*. Kluwer Academic Publ., Dordrecht.
- Kurek J., 1985. The general state-space model for a two-dimensional linear digital systems, *IEEE Trans. Autom. Contr.* AC-30, 600-602.
- Roesser R. P., 1975. A discrete state-space model for linear image processing. *IEEE Trans. On Autom. Contr.*, AC-20, 1, 1-10.
- Twardy M., 2007. An LMI approach to checking stability of 2D positive systems, *Bull. Pol. Acad. Techn. Sci.* vol. 55, no.4, 385-393.
- Valcher M. E., 1997. On the initial stability and asymptotic behavior of 2D positive systems, *IEEE Trans. on Circuits and Systems – I*, vol. 44, no 7, 602-613.

AUTHORS BIOGRAPHY

TADEUSZ KACZOREK, 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 several

universities. 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 850 scientific papers. He supervised 66 PhD theses. He is Editor-in-Chief of Bulletin of Polish Academy of Sciences, Technology Sciences and Editorial Member of about ten international journals.

DESIGN OF AUTOMOTIVE COMPONENTS USING ADVANCED CAE SYSTEMS

L. Barbieri^(a), F. Bruno^(b), M. L. Luchi^(c), G. Malito^(d), M. Muzzupappa^(e), F. Rotini^(f)

^(a,b,c,d,e)Department of Mechanical Engineering, University of Calabria, Rende (CS), Italy
^(f)Department of Mechanics and Industrial Technologies, University of Firenze, Firenze, Italy

^(a)loris.barbieri@unical.it, ^(b)f.bruno@unical.it, ^(c)luchi@unical.it, ^(d)gjorgio.malito@unical.it,
^(e)muzzupappa@unical.it, ^(f)federico.rotini@unifi.it

ABSTRACT

Simulation and other computer aided tools are often used in automotive design, since the design process is strictly oriented to the optimization of the performances through an iterative synthesis and analysis cycle aimed to understand the effects of changes in the geometry and layout of the various components. The search for the optimal performances is at the moment carried out empirically by the “trial and error” approach because parameters and constraints are too many for a global optimization of the vehicle dynamics to be performed. Nevertheless it is possible to introduce in the current design process some optimization algorithms or tools that can guide the designer in the decision process. This paper presents a methodology, applied to the conceptual design of the upright for a Formula SAE prototype, in which a multi-body simulator, CAD and a topological optimization tool are synergically employed in order to support the suspension design.

Keywords: multi-body, topological optimization, integrated approach

1. INTRODUCTION

The design of an automotive component has always been a complex process strictly dependent on the functions that the component must perform within the vehicle system. Therefore the final solution is almost never obtained directly but it is rather the result of an iterated optimization process which incorporates the capacity, the intuition, the experience and the know-how of the designer (Smith 1978; Staniforth 2006; Milliken and Milliken 1995).

The “build and test” method, historically adopted in engineering, is until now the only one that can be used to solve those problems which are not supported by suitable design instruments. The “build and test” method requires a physical prototype to be constructed and the boundary conditions in which it operates to be reproduced. This approach allows to find results that although accurate, are strictly related to the lay out of the experiment; it is hence practically impossible to gain any knowledge insight. Moreover, repeating the experiment is very onerous given of its high costs and

of the long time required. For these reason the build and test method is mostly adopted for the final validation of the results of design methodologies, rather than as a tool to search for the optimal solution.

The disadvantages of this method have been overcome by the development of CAE systems that allow to create digital mock-ups of the prototypes and to test them in the operating conditions in a completely virtual way. This latter approach allows to make changes at high speed and low cost so that the trial and error method becomes an extremely powerful design approach. The trial and error method in the case of multi-domain and complex problems, such as vehicle design problems, often involves the use of different CAx systems that, if utilized in an integrated manner, allow to reduce the time for developing the process and to achieve a high efficiency in comparison with the traditional methods.

The development process of a vehicle (fig.1) consists of four main phases starting from the definition of the vehicle specifications and ending with the identification of the characteristics required to the singular component.

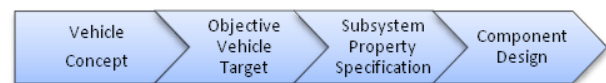


Figure 1: Vehicle development process.

In the first phase the vehicle concept is defined by identifying the characteristics and functionalities that the designer wants it to possess. In the next phase, the functional and performance macro-requirements, established in the previous phase, are objectified. In the third phase, the characteristics of the sub-systems are defined, taking into account the objective characteristics of the entire vehicle.

During the last phase the design process is utterly refined focusing on the single components of the different sub-systems.

It is hence possible to define three design levels (component, sub-system, vehicle) for each of which an iterative design process is identified within an analogous higher level design process; every level sends

inputs to and receives feedbacks from the upper level (fig.2).

The design of the upright is performed in the design process of the suspension subsystem and, indirectly, in the design process of the whole vehicle.

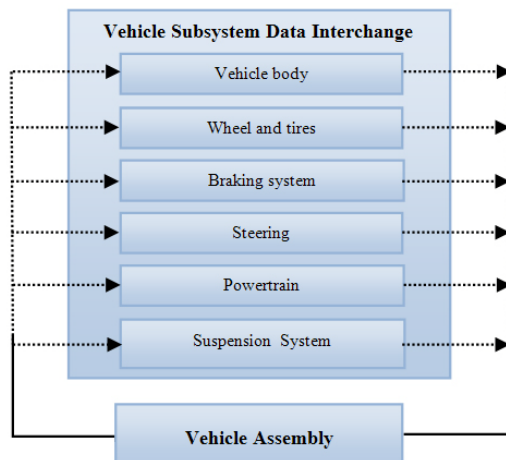


Figure 2: Chart flow subsystem design.

This paper presents an improved approach for an interdisciplinary combination of multi-body, CAD and topological optimization systems. In particular, the paper deals with the integration of the multi-body ADAMS/Car system with SolidWorks CAD system and Altair Optistruct topological optimization system, in order to achieve the optimal design of the upright of a Formula SAE prototype.

2. SUSPENSION COMPONENT DESIGN: THE STANDARD METHODOLOGY

The identification of requirements and constraints for the upright design descends from the dynamic behavior analysis of the whole vehicle. The ride and handling analyses allow to define design parameters for all components (damping, springs and structural suspension elements). In particular the handling analysis allows to determine the K&C suspension characteristics that are the basis for the structural component design. The suspensions are modeled generally in *multi-body* environments through kinematic constraints and shapeless elements, depending on the appointed architecture.

The use of these environments enable the designer to submit the system to K&C and optimization analyses in order to find a topology of the system that satisfies both the kinematic requests and the component compliance according to the elasticity requirements of the suspension.

The topological information are used in CAD system in conjunction with the data inferred from multi-body kinematic analysis in order to carry out a packaging analysis. Vice versa, the definition of the geometry of the components leads the kinematic optimization in MB system. Information concerning the

component compliance are utilized in FEM systems in order to carry out the structural test. Depending on the success or unsuccess of the structural test, these results are transferred to multi-body systems to either support or correct the values of the compliance.

Data exchange occurs between CAD and FEM systems: the geometrical modelling systems allow to define a geometry of the components compatible with packaging analysis, while structural simulators pick out a geometry that satisfies structural requirements. This iterative data exchange between MBS, FEM and CAD systems enables the optimal solution to be found for the entire system (fig.3).

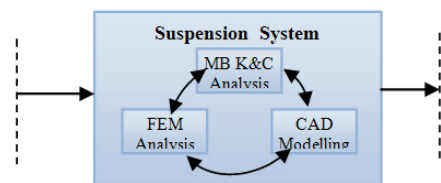


Figure 3: Vehicle suspension design.

2.1. Upright design

Design requirements and constraints of each component ensue from the analysis of the specific vehicle sub-system to which the component belongs (fig.4). Depending on the functionalities that the upright has to execute in the suspension sub-system, the requested properties are:

- An assembling geometry and a kinematic behavior into a predefined volume
- Specific compliance
- Light-weight structure

The upright compliance requirement is defined by means of the compliance analysis of the suspension sub-system (the suspension must undergo a given deflection under specified loads).

The kinematic and packaging analyses allow to determine the volume necessary to achieve optimal kinematics, functional surfaces essential for assembly, and overall dimensions such to avoid interferences during the dry run. In this way it is possible to specify geometric constraints that determine the maximum dimensions and the initial shape of the component.

Usually the material of the component is defined a priori on the basis of know-how, survey of competitors and economic valuations. Similar evaluations allow to choose an appropriate factor of safety. The loads with their application points, values and directions are the last characteristic to be determined.

The final optimal result will be a component with the minimum mass that satisfies the specified requirements.

Establishing loads and optimal geometry is the critical point of this approach. In fact, without a physical prototype it is necessary to calculate accurately

the normal operation loads. As a consequence, a robust virtual modeling is necessary in the multi-body system in order to simulate the normal operation of the suspension.

Concerning structural optimization, the traditional trial and error method enables the designer to make subsequent and iterative modifications motivated by geometry, stiffness and safety requirements.

The rate of convergence and the convergence itself depend strictly on the designer experience; the time could be very long; and the result could not be the best one.

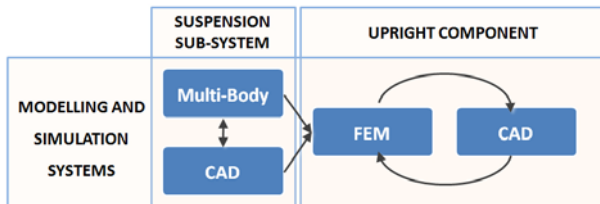


Figure 4: Specification and process for upright design.

3. SUSPENSION COMPONENT DESIGN: INTEGRATING TOPOLOGY OPTIMIZATION

As stated in the previous section, the use of a TO system during the development process (fig.4), allows to redefine the design methodology. In particular, the definition of the optimal geometry does no more depend only on the designer capabilities but it is supported by the three-dimensional topology optimization results. The integrated use of MB and TO systems allows to support more effectively the crucial points encountered during the design development process of an automotive component, that are: loads and optimal geometry definitions.

Figure 5 illustrates the proposed methodology. The methodology integrates three different systems and specifies the procedure for data exchanging.

As stated above, within the multi-body simulator the designer determines topological information, loads and constraints related to each component that has to be designed in detail. The topological information on the component (axis, distances, characteristic points, etc.) are used to define, through CAD system, the overall dimensions of the component geometry.

The use of TO systems simplifies the modeling phase because it is no longer necessary to specify the geometry in detail, but it is sufficient to define the component overall dimensions.

In the proposed approach, CAD automation tools have been adopted to improve the integration and interoperability between CAD and TO system (Barbieri et al. 2007; Barbieri et al. 2008; Barbieri et al. 2008) thus further simplifying the modelling phase. By adopting this tool the designer can specify in the model the invariant volumes which will not be modified during the optimization process. This knowledge can be stored and transferred directly to the next CAE phase to be reused in the redesign phase. This operative method offers several advantages. It allows to export the model

in CAE system no longer as a unique entity, but distinguishing invariant volumes from volumes to be optimized and, above all, it preserves this knowledge throughout the optimization process.

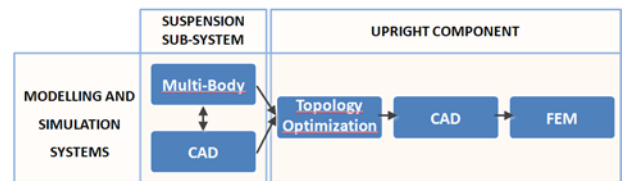


Figure 5: MB and TO integrated approach

In order to illustrate its capability, the approach is applied to the design of a FSAE upright.

4. CASE STUDY: FSAE UPRIGHT DESIGN

4.1. Problem definition

The vehicle taken as reference for the design of an upright is a prototype for Formula SAE competition realized in University of Calabria.

Following the development procedure previously described, in the first phase the fundamental characteristics of the vehicle have been defined, among them the suspension system double wishbone with pull rod for the front.

Subsequently, a first improvement of the set-up was achieved by a simplified model of the vehicle implemented in Matlab-Simulink. The model considers only the mass, the center of mass, the moments of inertia, the roll center, the geometric vehicle parameters and the tire characteristics. In this way the following parameters have been determined: wheelbase, track, mass distribution, suspension rates and suspension kinematics and compliance characteristics. The multi-body models of the suspensions and of the entire vehicle have been realized by the commercial software ADAMS/Car (fig.6).

During the maneuver operations of the vehicle, the adjustment of loads has been carried out by applying specific requirements on the spherical joints connecting the upright to the suspension. Values and directions of loads have been specified by giving the space coordinates relative to the coordinate system integral with the upright.

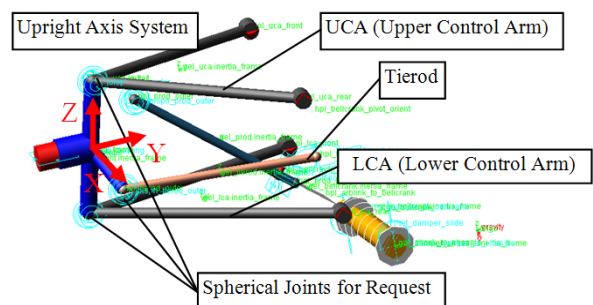


Figure 6: Multibody ADAMS\Car front-left suspension model.

4.2. Specifications for upright design

The stiffness requirements have been defined directly while modeling the set-up of the vehicle in Matlab-Simulink.

With regard to the material Aluminum 7075-T6 was selected because of its excellent ratio between weight and strength (Erickson, Heath, Woods, Dolan and Sears 2004; Jawad and Polega 2002; Jawad and Baumann 2002). The safety factor was chosen equal to 2.5 taking into consideration the contemporary presence of braking and cornering (Erickson, Heath, Woods, Dolan and Sears 2004; Jawad and Polega 2002; Jawad and Baumann 2002). When defining the functional surfaces, all the components which must be directly assembled with the upright have been considered (fig. 7). The dimensions of the wheel have been defined according to the pneumatic characteristics. These dimensions allow to determine the volume into which the upright has to be positioned. The selection of the bearing for the coupling with the hub and of its blocking system has permitted to establish the dimension of the seat of the bearing on the center of the upright.

The rod end bearing for the coupling with the lower control arm has determined the dimensions of the hole and of the slot located on the bottom part of the upright: the vertical distance between the opposite faces of the slot has been defined on the basis of the dimensions and of the mounting of the bearing; the space occupied during the kinematic movements of the suspension has dictated the horizontal distance between the opposite sides of the slot; the location of the hole has been established by kinematic analyses.

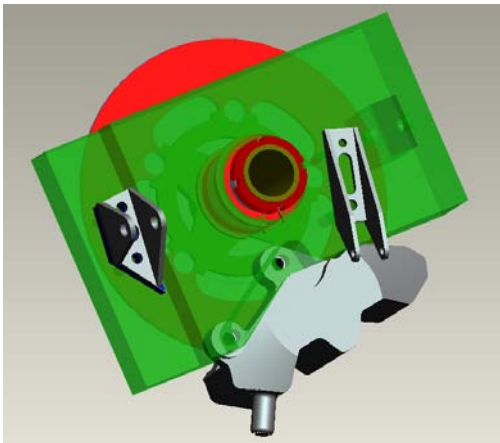


Figure 7: Upright and related sub-assembly

Because of the necessity, dictated by kinematics of shifting the rod end bearing, coupled with the upper control arm, towards the inner part of the vehicle and also to use plates for the camber regulation. A separate component, joined to the upright by two screws, has been added to accommodate the bearing seat. Starting from the dimensions of this new component, the surface and the position of holes for its assembly have been defined. A similar solution has been adopted to join the upright to the tie rod; the holes for its assembly

connection have also been identified. The selection of the brake gripper has permitted to define the holes and the surface for the assembly.

The optimal geometry of the upright will be found in the next TO phase; at this stage the CAD upright model is defined in a “rough” manner. In fact, during this first phase, the model must contain information about overall dimensions and the functional surfaces, resulting from the packaging analysis, that will remain unchanged during optimization.

To support these modelling operations a tool has been developed in SolidWorks which allows to specify and distinguish the invariant volumes from the volumes to be optimized. In this way the feature-based models of the invariant volumes can be transferred and reused in the redesign phase thus avoiding remodelling them (Barbieri et al. 2007; Barbieri et al. 2008; Barbieri et al. 2008).

4.3. Determination of dynamic loads

The loads to adopt in the design of the upright have been defined making reference to the worst conditions in relation with the maximum lateral and longitudinal accelerations recorded during standard maneuvers in a Formula SAE race.

Dealing with a frontal upright, braking and cornering simulations have been performed considering the combination of the two as the worst operative condition (Jawad and Polega 2002).

The loads taken as references in the analysis refer to the wheel mostly overloaded. The loads are given through their components in the SAE coordinate reference system of the upright and they are specified on the three points where the upright is connected to the tie rod, the upper control arm and the lower control arm.

Where cornering was concerned the model of the vehicle has traveled along a 30 meters diameter circular trajectory, at a reduced longitudinal acceleration, starting from the first gear and changing up to the limit of the cornering.

The braking maneuver has started from a velocity of 100 Km/h, in fourth gear, and jamming on the brakes until the complete stop of the vehicle.

For sake of brevity, only some of the charts of the analysis results are presented on the following (fig. 8-9):

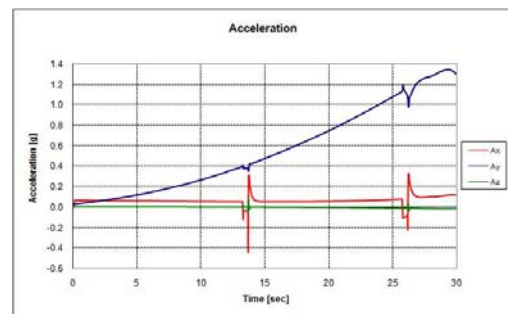


Figure 8: Chassis acceleration during cornering analysis (Vehicle Axis System).

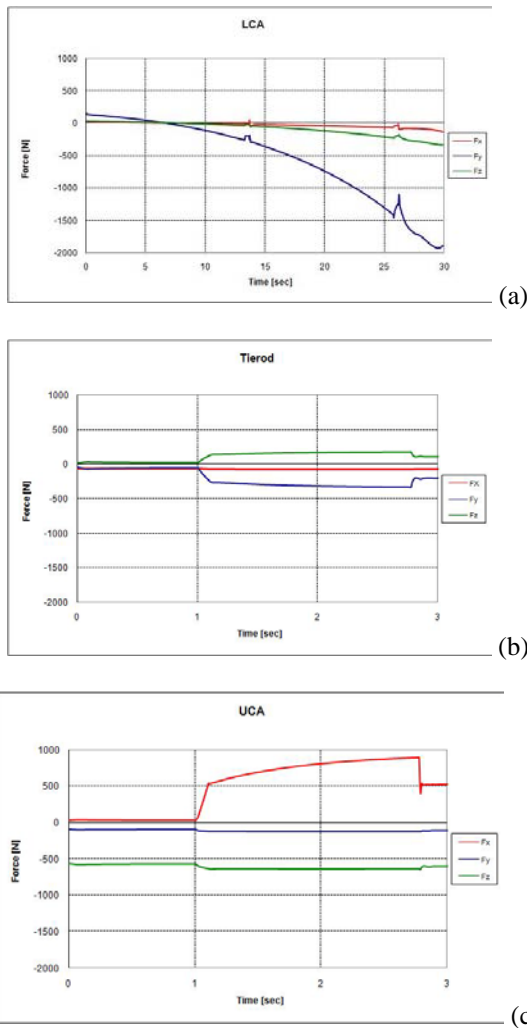


Figure 9: Time-history (a) LCA loads during cornering analysis, (b) Tierod and (c) UCA loads during braking analysis (Upright Axis System).

The maximum upright load values are shown in table 1.

Table 1: Upright load values
Braking and cornering combination loads value

	Fx [N]	Fy [N]	Fz [N]
UCA	936.6	1357.5	-765.7
LCA	-1620.0	-1926.9	-341.3
TCA	153.1	-539.9	279.2

4.4. Topology optimization analysis

The topological optimization analysis was performed using Altair Optistruct. Before starting the analysis, it is necessary to create the mesh of the model. Altair offers, in the Geometry panel, various tools that allow the designer to create the model but, as stated before, it is possible to import the upright surfaces in IGES format. In this phase, the designer has simply to define the collectors, that is to declare the physical properties of the model and the boundary conditions identified during multi-body analysis.

It is possible to create a collector for specifying the type of material, another for loads and constraints, other

ones to define variant and/or invariant volumes. The import of invariant volumes from CAD system facilitates the operation of selection of hexahedral elements assigned to “no design” mesh (fig.10).

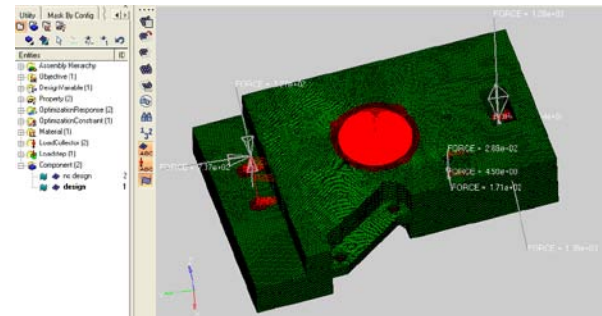


Figure 10: Mesh generation and loads and constraints assignment

The topology optimization problem is then defined as follows: minimize the volume of the upright.

At the end of the optimization process it is necessary to execute the OssSmooth module, present in Altair Hyper Works, in order to translate the topological optimization results into an STL format file and export it in CAD environment (fig.11).

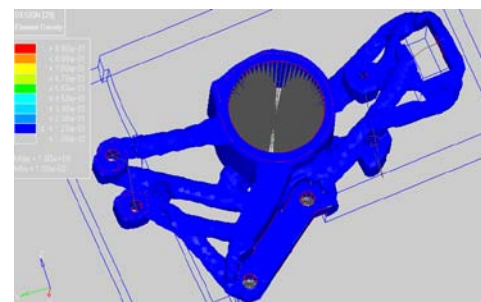


Figure 11: Topology optimization results

4.5. Model redesign

In this phase, the result of the optimization is imported into a CAD environment for modeling the optimized feature-based upright. At present, the translation of optimization results into a feature-based model is not an operation adequately supported by CAD systems. One of the main problems regards the loss of all feature data in the optimization process. The output of topological optimizers is a voxel model, so it does not take into account either functional and technological features, either rules and attributes associated to the various parts of the product. There are many interpretation techniques available in the literature but most of them focus on 2D or 3D simplified structural optimization applications (Hsu, Hsu and Chen 2001; Hsu and Hsu 2005).

In order to simplify this phase, a SolidWorks macro was implemented that allows to automatically input invariant geometries identified during the initial phases of the procedure.

In future, during this phase, we will introduce the geometric analysis for investigating topologically

optimized geometries in order to reduce user interactions and decision-makings in the solid model reconstruction (Barbieri et al. 2008, Barbieri et al. 2008). In figure 12 the final upright model is shown.

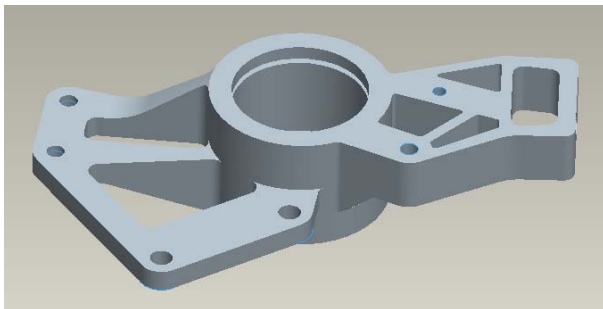


Figure 12: Final feature-based model

4.6. Finite element analysis

In the last phase of the proposed methodology, classical FE approaches are applied to test and validate the new shape of the optimized model (fig.13).

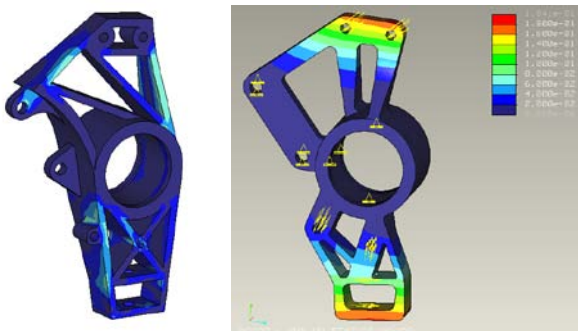


Figure 13: Comparison among FEM model of the upright 2008 and the new upright.

The comparison, between this model and the upright currently present in the FSAE car, shows a significant improvement in term of weight and stiffness (Table 2).

Table 2: Data comparison

	Upright 2008	Upright 2009
Maximum Stress (Mpa)	94.2	168
Maximum Deflection (mm)	0.24	0.18
Weight (Kg)	0.966	0.721

5. CONCLUSION

In the paper a methodology has been proposed in which multi-body simulation, CAD and topological optimization are synergically employed to support the suspension design. The methodology has been applied to the conceptual design of the Upright for a Formula SAE prototype.

The results have shown that the integrated design approach can be an efficient support in the optimum conceptual design of a mechanical component with

complex dynamic behavior, particularly when almost no experience on the system is available. The comparison between FSAE Upright version 2008 (designed by the traditional approach), and FSAE Upright version 2009 (designed by the proposed approach) show a significant improvement of the mechanical properties.

REFERENCES

- Barbieri, L., et al., 2007. CAD automation and KB user interfaces for an efficient integration of topological optimization tools in the product development process, *4th Int. Conf. on PLM*, Bergamo, Italy.
- Barbieri, L., et al., 2008. Design Automation tools as a support for knowledge management in topology optimization, *Proceeding of IDETC08*, New York.
- Barbieri, L., et al., 2008. Guidelines for an efficient integration of Computer Aided Innovation with optimization and plm systems, *2nd European HyperWorks Tech. Conf. (HTC08)*. Strasbourg, Germany.
- Erickson, G.W., Heath, M.R., Woods, B., Dolan, D.F., Sears, J.W., 2004. Design and Manufacture of Titanium Formula SAE Uprights using Laser-Powder-Deposition, *SAE Paper N. 2004-01-3546*.
- Hsu, Y.L., Hsu, M.S., Chen, C.T., 2001. Interpreting results from topology optimization using density contours, *Computers and Structures*, 79:1049-58.
- Hsu, Y.L. and Hsu, M.S., 2005. Interpreting three-dimensional structural topology optimization results, *Computers and Structures*, 83:327-337.
- Jawad, B. A. and Baumann, J., 2002. Design of Formula SAE Suspension, *SAE Paper N. 2002-01-3310*.
- Jawad, B. A. and Polega, D. B., 2002. Design of Formula SAE Suspension Components, *SAE Paper N. 2002-01-3308*.
- Milliken, W. F. and D. L. Milliken, 1995. *Race Car Vehicle Dynamics*, SAE Publishing.
- Smith, C., 1978. *Tune to Win*, Aero Publishers, Inc. Distributed by SAE.
- Staniforth, A., 2006. *Competition Car Suspension: A practical handbook*, Haynes Publishing.

AUTHORS BIOGRAPHY

Loris Barbieri: PhD student in Mechanical Engineering at University of Calabria. He received a Master's degree in Mechanical Engineering at University of Calabria in 2006. His research focus on implementation of methods and tools to improve the integration of advanced and innovative instruments of design, such as VR and topological optimization systems, in PDP; reverse engineering; KBE systems and Computer Graphics.

Fabio Bruno: He is an Assistant Professor at the Department of Mechanical Engineering of the University of Calabria. He received a Master's degree in

Industrial Engineering in 2001. In 2006 he obtained a PhD in Mechanical Engineering at University of Calabria. The research activity mainly concerns the application of Virtual and Augmented Reality (VR & AR) to the design process of the industrial products.

Maria Laura Luchi: Full professor of Engineering Drawing at the Department of Mechanical Engineering of the Faculty of Engineering of the University of Calabria. Dean of the Faculty of Engineering since 1997 to 2008. Her research activities have been addressed to the numerical methods for stress analysis (particularly the finite elements methods and the boundary elements methods applied to fracture mechanics and to ceramic materials), to solid modelling system (particularly the features-based system), to the image processing (particularly the images obtained by interferometric techniques) and to web-based design.

Giorgio Malito: He is a PhD Student in Mechanical Engineering at University of Calabria. He received a Master's degree in Mechanical Engineering in 2006. His research focus on multibody systems and vehicle dynamics.

Maurizio Muzzupappa: He 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.

Federico Rotini: he took his degree in Mechanical Engineering at the University of Florence 2001, discussing the thesis "A Semantic Knowledge portal to assist the design of plastic parts". He is Researcher in Methods and Tools for Machine Design since 2002 at Florence University, and he has become a Senior Researcher in 2005. His main fields of activity are methods and techniques for Systematic Innovation, integration of CAD/CAE tools, Optimization techniques, integration among CAI tools and Optimization tools.

DEVELOPMENT OF A CARGO SCREENING PROCESS SIMULATOR: A FIRST APPROACH

Peer-Olaf Siebers ^(a), Galina Sherman ^(b), Uwe Aickelin ^(c)

^(a) Computer Science, Nottingham University, Nottingham, NG8 1BB, UK

^(b) CASS Business School, City University London, London, EC1Y 8TZ, UK

^(c) Computer Science, Nottingham University, Nottingham, NG8 1BB, UK

^(a) pos@cs.nott.ac.uk, ^(b) galina.sherman.1@cass.city.ac.uk, ^(c) uxa@cs.nott.ac.uk

ABSTRACT

The efficiency of current cargo screening processes at sea and air ports is largely unknown as few benchmarks exist against which they could be measured. Some manufacturers provide benchmarks for individual sensors but we found no benchmarks that take a holistic view of the overall screening procedures and no benchmarks that take operator variability into account. Just adding up resources and manpower used is not an effective way for assessing systems where human decision-making and operator compliance to rules play a vital role. Our aim is to develop a decision support tool (cargo-screening system simulator) that will map the right technology and manpower to the right commodity-threat combination in order to maximise detection rates. In this paper we present our ideas for developing such a system and highlight the research challenges we have identified. Then we introduce our first case study and report on the progress we have made so far.

Keywords: port security, cargo screening, modelling and simulation, decision support, detection rate matrix

1. INTRODUCTION

The primary goal of cargo screening at sea ports and air ports is to detect human stowaways, conventional, nuclear, chemical and radiological weapons and other potential threats. This is an extremely difficult task due to the sheer volume of cargo being moved through ports between countries. For example in sea freight, 200 million containers are moved through 220 ports around the globe every year; this is 90% of all non bulk sea cargo (Dorndorf, Herbers, Panascia, and Zimmermann 2007).

Little is known about the efficiency of current cargo screening processes as few benchmarks exist against which they could be measured (e.g. %detected vs. %missed). Some manufacturer benchmarks are available for individual sensors, but these have been measured under laboratory conditions. It is rare to find unbiased benchmarks that come from independent field tests under real world conditions. Furthermore, we have not found any benchmarks that take a holistic view of

the entire screening process assessing a combination of sensors and also taking operator skills, judgment and variability into account.

In our research we attempt to identify and test innovative methods in order to advance the use of simulation for supporting decision making at the strategic and the operational level of the cargo screening process. Wilson (2005) confirms the usefulness of simulation for the analysis and prediction of operational effectiveness, efficiency, and detection rates of existing or proposed security systems.

Our research aim is to develop a methodology for building such Decision Support Systems (DSS) that will map the right technology and manpower to the right commodity-threat combination in order to maximise detection rates. The concept for such a DSS (a cargo screening process simulator) is shown in Figure 1. For developing the methodology we are using a case study approach. In our work we focus solely on DSSs development; we do not work on new sensor development. However, with our DSSs we might be able to give some recommendations of what characteristics new to be developed sensors might require to reach certain system performances.

The core of the proposed cargo screening process simulator will consist of three elements: a Detection Rate Matrix (DRM), a simulation model and a resource optimiser. The DRM will provide sensor detection rates as an input for the sensors represented in the simulation model, based on sensor types, commodities, threats, and other indicators. The simulation model will allow carrying out what-if analyses for the system under examination. The results of the simulation will be fed into the resource optimiser to create a new set of input parameter values for the simulation. The previous two steps are repeated until an acceptable solution has been found. The output of the simulator will consist of required technology and manpower and an estimation of the system detection rate that can be achieved by implementing the proposed system set-up. A sensor data database will provide some information for the core elements (in particular for the DRM). The content of the database will be a mixture of data provided by vendors but will also consider operators experience with

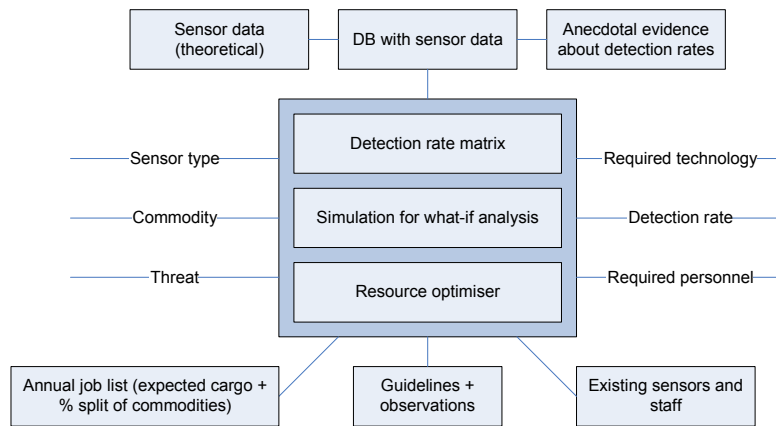


Figure 1: Conceptual Model of Our Cargo Screening Process Simulator

the equipment. Other input data required for the cargo screening process simulator include an annual job list, guideline on how to carry out jobs, and observations if jobs are carried out in accordance with these guidelines, and a list of existing sensors and staff.

Section 2 contains a brief review of existing work in the field. In Section 3 we discuss the development of a DRM. In Section 4 we state our research questions regarding model design and matrix development. Section 5 introduces our case study, the ferry port in Calais. We present a description of the real system and its operations, a conceptual model of it, an implementation of the conceptual model in form of a discrete event simulation model, and finally we show the results of an initial test run with our simulation model. Section 6 concludes the paper by discussing the results of our current efforts and proposes further work.

2. BACKGROUND

Simulation modelling is commonly used to support design and analysis of complex systems. With regards to modelling ports Tahar and Hussain (2000) confirm that simulation modelling is a tool widely used for the management, planning and optimisation of port systems. According to Turner and Williams (2005) the same is true for the management, planning and optimisation of complex supply chain systems.

In the context of the cargo screening process, some examples (e.g. Leone and Liu 2005, Wilson 2005) have been found that use simulation modelling to evaluate key design parameters for checked baggage security screening systems in airports, in order to balance equipment cost, passenger and baggage demand, screening capacity, and security effectiveness in an attempt to meet the requirements imposed by the checked baggage screening explosive detection deadline established by the US Aviation and Transportation Security Act.

Another related subject is the enhancement of the security throughout the supply chain, i.e. achieving supply chain integrity (Closs and McGarrell 2004). Here, simulation modelling is often used to analyse the system. For example, Sekine, Campos-Náñez, Harrald, and Abeledo (2006) use simulation and the response

surface method for a trade-off analysis of port security in order to construct a set of Pareto optimal solutions. The development of a dynamic security airport simulation is described by Weiss (2008). In contrast to the other papers mentioned so far, this simulation focuses on the human aspects in the system and employs the agent paradigm to represent the behaviour of attackers and defenders. Both, attacker and defender agents are equipped with the capability to make their individual decisions after assessing the current situation and to adapt their general behaviour through learning from previous experiences. This allows accounting for rapid security adaptation to shifting threads, as they might be experienced in the real world.

3. CONCEPTS OF THE DETECTION RATE MATRIX

The mapping process (right technology and manpower to the right commodity-threat combination) will be implemented using a multi-dimensional DRM. The DRM contains the information required to estimate the type and amount of sensors and manpower we need in order to maximise our detection rate if we have an estimate of the number and type of cargo containers we want to screen and what they will contain. The values to fill the DRM can either come from vendors, the literature, from trials, or anecdotal evidence of the border agency staff. From all the information received we have to create a single value that represents the detection rate for a certain commodity-threat combination.

An example for a partially filled DRM derived from laboratory experiments can be found in Klock (2005). Klock states that developing a DRM from real world data would be desirable but poses a big challenge as for various reasons it is a problem to collect all relevant data for the all commodity-threat combinations in the real system. In our case the problems are as follows. In most cases the security screening procedures cannot be compromised for research purposes, i.e. there are legal boundaries regarding the sampling frame. Furthermore, it would be difficult to capture the variability of operational procedures that exist in the real system. However, as much as the technology itself,

the way in which the technology is used contributes to the success rate of detecting threats. Our current plan is to fill some of the gaps in our DRM by simulating specific scenarios, rather than trying to collect all data from the real system.

We will start the development of our DRM by creating a two dimensional matrix and then gradually increase its complexity (i.e. the number of dimensions). The values for our first DRM will be derived by collecting anecdotal evidence from system insiders and where anecdotal evidence is not available by simulating specific scenarios of interest (1). The next step will be to generalise the initial DRM and to consider that the applicability and performance of sensors is related to the commodity screened and the category of threats investigated (2). For example, if one wants to detect stowaways in a lorry using CO2 probes which measure the level of carbon dioxide and the load consist of wood or wooden furniture which naturally exhumes carbon dioxide then the detector readings will be wrong. For this commodity the sensor is not useful and would produce many false positives (type 1 error), which means that in return many false negatives (type 2 error) will stay undetected as time is wasted with manually inspecting the wrong lorries. The next dimension we will add is a definition of the cargo containment which consists of a description of the type of containment, its wall thickness and its wall density (3). The containment type is important as some of the sensors might need to have access to the interior of the containment while others might be applicable to be used from the outside. Wall thickness and density are important as many sensors have limitations regarding the penetration of the containments, depending on the containment properties.

$$\text{rate of detection} = f(\text{commodity \& thread combination, specific scenario}) \quad (1)$$

$$\text{rate of detection} = f(\text{commodity, threat, sensor}) \quad (2)$$

$$\text{rate of detection} = f(\text{cargo containment, commodity, threat, sensor}) \quad (3)$$

There are many more dimensions one could add (e.g. cargo origin, cargo destination, shipping company, or environmental conditions of test facility location) and part of the research will have to deal with the question of which are the most relevant indicators of sensor efficiency?

4. RESEARCH QUESTION

One of the key questions we are keen on answering during our research is how and where it makes sense to use simulation in a project like ours. Besides the standard application areas for simulation modelling in operations research (e.g. system analysis, optimisation, as a communication tool) we want to find and test some new application areas (e.g. validating the DRM parts where we have data and helping to estimate the values where we have gaps in our DRM). Furthermore, we will

examine if our simulation models can be used to support the decision making process in other fields, e.g. supply chain management or risk analysis.

Before we can build our cargo screening process simulator we will have to investigate several questions which can be broadly grouped in two categories, related to model design or matrix development. Research questions regarding model design: [a] How much detail do we have to model to get some meaningful output? [b] How should we model people in our system (e.g. officers or stowaways) - as simple resources or as autonomous entities? [c] How can we get a good estimate on how many stowaways, weapons or drugs are passing the borders? [d] What effect does the fact that we are dealing with rare events have on input sampling and output analysis? Research questions regarding matrix development: [e] Which are the most relevant indicators of sensor efficiency? [f] What is the best way to develop and validate a detection rate matrix in absence of real data or when real data is incomplete, i.e. missing data for certain technology / commodity / threat combinations? [g] Can we develop a framework to support the development of a DRM for different environments and for different threats?

5. CASE STUDY: CALAIS FERRY PORT

In order to achieve our research aim of developing a methodology for building cargo screening process simulator we have chosen to use a case study approach. This allows us to gain the knowledge, insight, and experience we need for developing our methodology. For each case study we will first develop simulation models that allow us to analyse the system under study and then create a DRM for this system.

For our first case study we have selected the ferry port in Calais (France) that links Calais with Dover (UK). This site is ideal for beginning as the security measures in place focus on detecting only one threat, illegal immigrants, or clandestines, as they are called by the UK Border Force. Clandestines are people found on a lorry with the aim to get into Britain without a passport or any other papers (Sky1 2009). These can be individuals or groups. Clandestines come in hope of a better future in Britain, drawn by the English language, the lack of national identity cards and the possibility of illegal work. When clandestines do not succeed little or no publicity is generated, thereby perpetuating the false idea that clandestines are always successful. On the other hand, for every successful clandestine arriving in Britain the word goes out that the process is successful, which generates even more attempts of illegal immigration (Brown 1995).

5.1. The Real System

Between April 2007 and April 2008 more than 900,000 lorries passed the check points in Calais. Of these, approx. 0.3% contained additional human freight (UK Border Agency 2009). How many clandestines were missed during these checks is unknown. Although companies supplying the sensor technology promise a

detection rate close to 100%, independent test have shown that this is not the case when using the equipment in real world scenarios (Klock 2005). In addition, in the real system the detection rates also depend on factors like the time of day (at busy times the operators have less time to apply the sensors and wait for the readings and therefore readings are more likely to produce more type I and type II errors), operators' skills (of interpreting the outputs from the sensors), and operators' fatigue.

In Calais the cargo screening process is separated into two major zones, the first under the control of the Calais Chamber of Commerce (CCI), the second under the control of the UK Border Agency (see Figure 2). Different types of sensors are used at the various screening facilities and some of them are also in use as mobile devices. The technology / operations used for screening includes Passive MilliMetre Wave scanners (PMMW), Heart Beat Detectors (HBD), CO2 measurement probes (CO2), canine sniffers and visual inspection. The process on the French site starts with a passport check by the French authorities. Then all lorries are screened for clandestines and suspicious lorries are routed to deep search facilities where they are further inspected by using an alternative method and if suspicion is substantiated then lorries are opened for visual inspection. In some cases (e.g. if it does not interrupt the process flow much, e.g. at non-busy times) lorries are opened directly for a quick visual check after or instead of being screened. If clandestines are found on board a lorry they are removed by the French police, registered, and released into freedom. The process on the UK site is very similar; the major difference is that lorries are searched rather than screened and that only a fraction of the lorries going through the system is actually searched (at average 33%). The number of vehicles searched is on the basis of profiling and intelligence. Once the lorries have passed all check points they park at the Berth where mobile squads are operating to check the lorries a last time before they get on their way to Dover.

5.2. Modelling the Real System: A First Approach

This initial modelling exercise acts as a data requirement analysis for our case study. It will help us to make informed decisions about the information and data we need to collect during our main data collection

for this case study. Furthermore, it will help us to uncover areas where we might encounter problems during our main modelling and implementation process at an early stage, so that we can respond to it in sufficient time. Finally, we want to use our initial models to communicate theories, ideas, potential investigation techniques, outputs and solutions to stake holders and other interested parties.

Before we started our modelling exercise we visited the case study site to observe the operations, for discussions with stake holders, and for collecting system performance data. From the information gathered we developed a conceptual model that reflects the current operations of the cargo screening process at the ferry port in Calais.

5.2.1. Modelling Challenges

The case study system presents several modelling challenges, some of which have already been mentioned in Section 4. Below is a list of the modelling challenges we are currently facing. The first challenge is related to the fact that we are dealing here with a complex system where factors that are difficult to quantify are assumed to have a big impact on system behaviour and ultimately system performance. An example for such a factor is the human decision making process. Therefore, the application of abstraction and simplification for the purpose of model design is a very delicate issue.

The second challenge is related to the lack of input data. On the one hand we are dealing with rare events (e.g. detecting a clandestine) which impacts on the way we have to do our input sampling and output analysis (Heidelberger 1995) and some data cannot be obtained from the real system (e.g. number of clandestines that manage to cross the borders) so we have to make a lot of guesses. Some mathematical models exist for estimate such values as for example success rates for clandestine border crossing (Wein, Liu, and Motskin 2009; Epenshade 1995); their usefulness however is debatable as still many assumptions have to be made to derive these estimates. Even if we had the resources to collect the data there are some legal issues regarding the sampling frame which prohibits us to collect some of the required data as we are not allowed to sample an entire population.

The third challenge relates to the objects we have to model, some of which are fixed and some of which

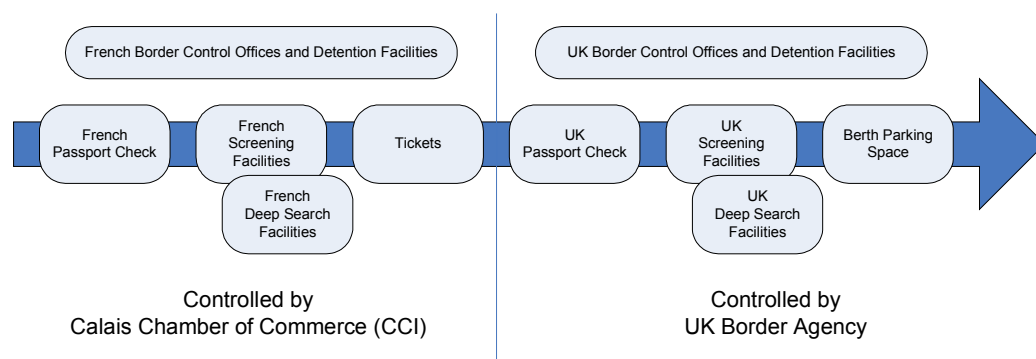


Figure 2: The cargo screening process at Calais

can be moving around freely. Three different situations can be identified: [1] Sensors are fixed and targets (lorries and clandestines) are fixed, for example in a screening shed lorries are parking while sensors are applied. [2] Sensors are moving and targets (clandestines) are moving, for example in the allocation lanes officers are patrolling and clandestines are running around in order to get into the lorries. [3] Sensors are moving and targets (lorries and clandestines) are fixed, for example in the Berth squads are checking the parking lorries either with mobile sensors or by opening suspicious lorries directly. While the first situation is relatively easy to model using traditional Discrete Event Modelling (DEM) the latter two require some further reflection before they can be modelled successfully. In those cases sensors and/or targets need to possess some form of autonomy and probably proactiveness which are concepts not directly supported by traditional DEM. Agent-Based Modelling (ABM) presents an alternative modelling paradigm that supports the consideration of autonomy and proactiveness of entities.

The fourth challenge relates to the injection of clandestines into the model. Anecdotal evidence suggests that clandestines enter the system at numerous places. Clandestines either get into the lorries before these enter the compound or they climb over the fences that surround the compound and get into the lorries while they pass through the compound or wait for the ferry. While the first is easy to model the latter causes some problems as no data is available exactly where and when the clandestines enter the compound.

Finally, the fifth challenge relates to the human decision making. The operation of this system is human centric and relies very much on the experience of the officers and the compliance to rules. Human decision making involves the routing (i.e. choosing the lorries to be screened), choosing the sensor to be used, interpreting the sensor outputs (i.e. choosing the lorries to be opened), and compliance to rules (sticking to recommended sensor application periods). All these points depend very much on the state of the system. At peak times decisions will be different compared to quiet times, e.g. sensor application periods will be shorter to avoid congestion in front of the service sheds and therefore the number of true and false negatives will be much higher and therefore the detection rates vary throughout the day.

In the end the big question is if modelling all these details is really necessary for getting useful results. In order to answer this question we will have to implement them at least partially and conduct a sensitivity analysis. For this purpose we will build some smaller simulation models that only represent a small section of the overall real system. Once we have the results we can give some recommendations regarding the level of detail that is required for getting a useful representation of the real system.

5.2.2. The Conceptual Model

In order to capture the cargo screening process taking place at the Calais ferry port we have developed a process centric conceptual model (Figure 3). It reflects the process flow as it appears in the real system. Dark blue fields represent system entry and exit points. Brown fields represent jump starting points (where the text is followed by @) and targets (where the text is lead by @). These jumps do not consume any time. Light blue fields represent the locations where time is consumed. The %s represent flow probabilities while the numbers below the light blue fields represent detection rates. For confidentiality reasons the true values have been replaced by place holders. The splits in the model have been defined in a somewhat arbitrary way but often a single row represents all activities that happen at one specific location.

5.3. Implementation

Based on our conceptual model presented in Section 5.2.2 we have developed a first version of a Discrete Event Simulation (DES) model which is implemented in AnyLogic Version 6.4, a java-based multi-paradigm simulation software. The purpose of this exercise is to identify where we have gaps in knowledge about the system and to identify missing data that could be obtained during our main data collection.

5.3.1. Simulation Software

The object-oriented model design paradigm supported by AnyLogic provides for modular, hierarchical, and incremental construction of large models (XJ Technologies 2009). Each model contains a set of active objects which often represent objects found in the real world. At the lowest level these active objects can contain parameters, variables, functions, events, state charts and other active objects. For DES modelling there is also a library containing higher-level objects that support the creation of discrete event patterns frequently used in process-centric modelling (e.g. entity generation, buffering, resource usage, entity routing, entity destruction).

One of the benefits of AnyLogic is that you can build mixed models, i.e. you can mix process-centric DEM and individual based ABM in one hybrid simulation model. Technically the main difference is that an agent compared to an active object has some additional features with respect to dynamic creation and destruction, synchronisation, space-, mobility-, and spatial animation, agent connections and agent communication. We will use these features when we model for example sensor and target movement in the allocation lanes.

For our current simulation model we use the elements from the library but we have also developed our own element in form of embedded active objects that contain a collection of parameters, variables and library elements. These are reusable components that can currently represent any of the service sheds as well as passport and ticket booth on the Calais compound.

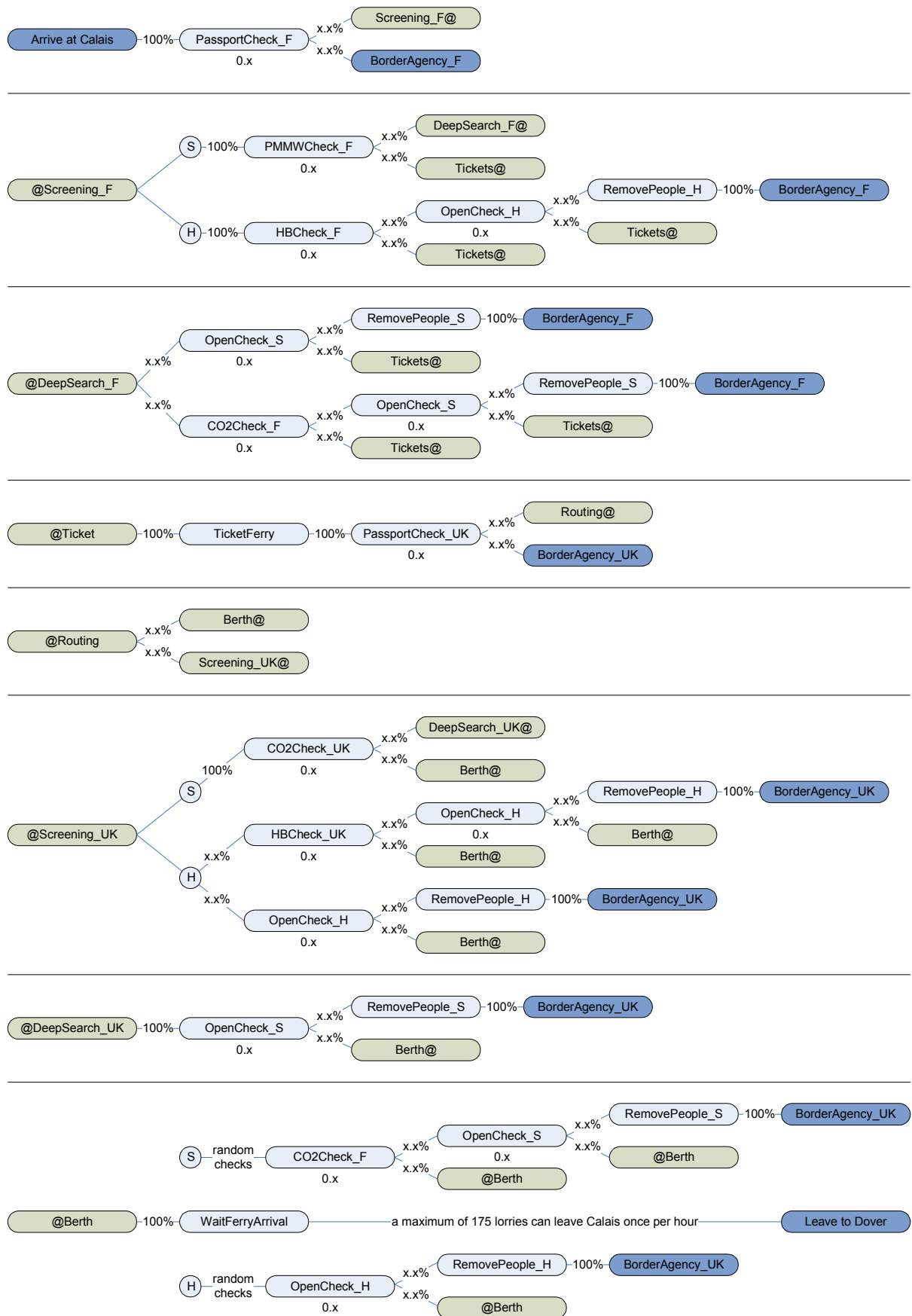


Figure 3: Conceptual model of the cargo screening process at Calais (S=soft sided, H=hard sided, F=French site; UK=UK site; x=replacement for real value)

We tried to make them as generic as possible so that we can also use them for modelling other locations or types of systems. A screenshot of the elements of such an embedded active object is presented in Figure 4. It shows a service station with a linked resource pool (symbolised by the clock and the linked box) an entrance buffer and two single space exit buffers. The hold element between entrance queue and service station is released to let one entity pass at a time as soon as the previously serviced entity has left the exit buffer, which only happens if there is some space available in one of the upstream queues. The variables on the left are used for data collection while the parameters on the right allow each instance of this active object class to be defined by an individual set of parameters.

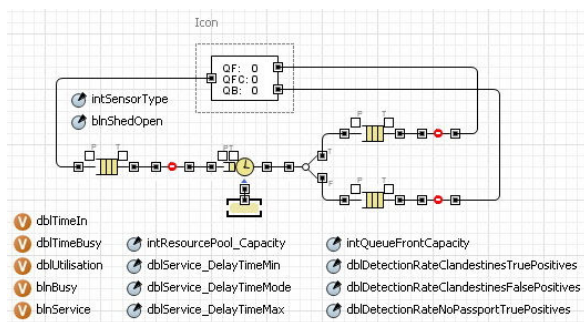


Figure 4: Embedded active object serviceShed

5.3.2. Simulation Model

Entities of type Lorry (soft and hard sided) are injected into the simulation at a certain rate by the source element (arrival). Some lorries will arrive with an additional load (clandestines) on board. Clandestines are currently modelled as resources (a boolean variable defines if there are clandestines on board a lorry or not). The main elements in the simulation model are the serviceShed elements which have been described in Section 5.3.1. These are used for modelling the situations when we have fixed sensors and fixed targets. The serviceShed elements are linked via some routing elements. The routing elements use a custom-made function which routes the entity to the next level upstream element with the shortest queue. This represents the routing activities normally conducted by an officer.

Figure 5 displays a section of the simulation model within the AnyLogic IDE. The project view window on the left shows the project tree of the current project. The graphical editor in the middle shows the content of the Main object. The pallet window on the right displays the different pallets available in AnyLogic, amongst them the Enterprise Library pallet. The properties window at the bottom is used to define the properties of the element, which can contain Java commands and method calls.

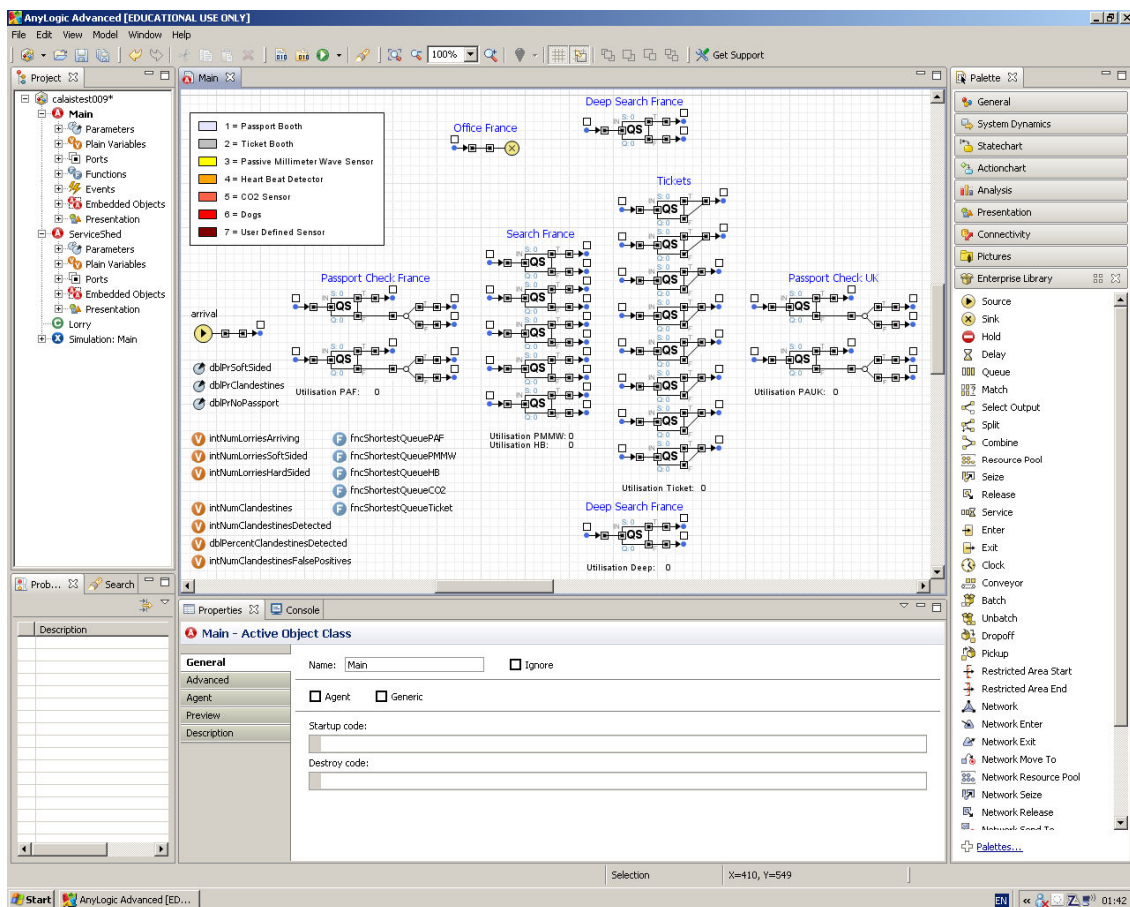


Figure 5: A section of the simulation model within the AnyLogic IDE

For modelling the Berth activities we could not use our serviceShed elements as we have a situation with moving sensors and fixed targets. Instead we developed the solution presented in Figure 6. The moving squads are modelled by events that pick one lorry at random (mobile CO2 checks will be conducted on soft sided lorries while hard sided lorries will be opened) and check it. The time it takes to check a lorry is represented by the inter arrival time between two events. This means that the squads are currently modelled as being 100% utilised as long as there are lorries to check. There are two modus operandi, either lorries can be checked only ones (lorries that have been checked already are registered on an ignore list) or lorries can be checked multiple times (which represents the situation where clandestines enter the lorries while these are parking at the Berth and therefore the squad would check suspicious lorries again).

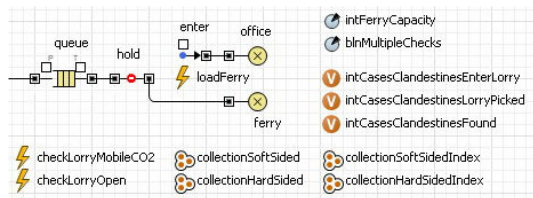


Figure 6: Modelling the Berth activities

To set up the model parameters we had to use some best guesses and common sense. However, as we have some average input and output data we have tried to use settings for the unknown parameters to match the output data of the real system when using the input data of the real system. A problem is that the available data are average data for multiple stations or monthly averages. Where ever possible we have used multiple data sources for estimating values for the data required (e.g. for calculating process flow probabilities).

5.3.3. Current Omissions

This simulation model presented here represents the first draft of our aspired DES model. The main simplifications and abstractions in the current model are listed below:

- Queues: We use large queue capacities in front of the service stations; therefore congestion does not occur. However, it is an important phenomenon that occurs in the real system and influences service times and therefore indirectly the detection rates.
- Average values: We use the same average entity arrival rates for the entire simulation runtime but the collected data indicates a significant difference in arrival rates as well as inspection and detection rates depending on time of the day and day of the week (the higher the arrival rates the lower detection rates, as officers have less time for conducting an individual screening). However, first we need

to sort out the congestion problem mentioned above; otherwise the impact of high arrival rates is not adequately considered in the results.

- Currently we don't model multiple clandestine entry points, canine sniffers nor the search for clandestines in the allocation lanes.

5.4. Testing the Simulation Model

So far we have only conducted some very basic preliminary tests with our simulation model. A verification and validation exercise is still to be carried out. However, here we briefly report on one of the tests we have conducted. We have set up the simulation model using our standard set of parameter values, except for the sensor detection rates, which we have set to the same value for all sensors. During the experiment we have systematically changed this collective value, starting from 0% to 100% in steps of 10%. Our simulated runtime was equivalent to a one year period and we conducted 20 replications for each set of values. As for the results we expect to see a non-linear relationship between sensor detection rates and the average proportions of clandestines detected. This is due to the fact that many lorries will go through several screening procedures and therefore combinatorial effects appears for this relationship, where higher individual sensor detection rates will have a proportionally lower benefit regarding the system detection rate. Figure 7 confirms our expectation.

This first test has already shown the impact of modelling rare events. We observed that the clandestine detection rates vary significantly throughout most of the simulation runtime and seems only to stabilise towards the end. Furthermore, we noticed some significant differences between runs. Therefore, in future we have to assess very carefully the required warm-up period, run length and number of replications.

6. CONCLUSIONS

In this paper we have presented our first steps towards the development of a cargo-screening process simulator and we have introduced a case study that we want to use for gaining some experience with developing such a simulator. Our current task is to conduct a data requirement analysis. For this we have created a first draft of our aspired DES model to be used for the cargo-screening process simulator. This modelling exercise has allowed us to make a well informed decision about which kind of information and data we require for representing the real system to allow some useful systems analysis.

We found that a big challenge when modelling the case study system is to capture the variability inherent in the system. By omitting details like differences in arrival rates throughout the day and week, congestion in front of service sheds and associated with this service time variation and detection rate variations we do not get a good representation of the real system, in particular when we are not only interested in the

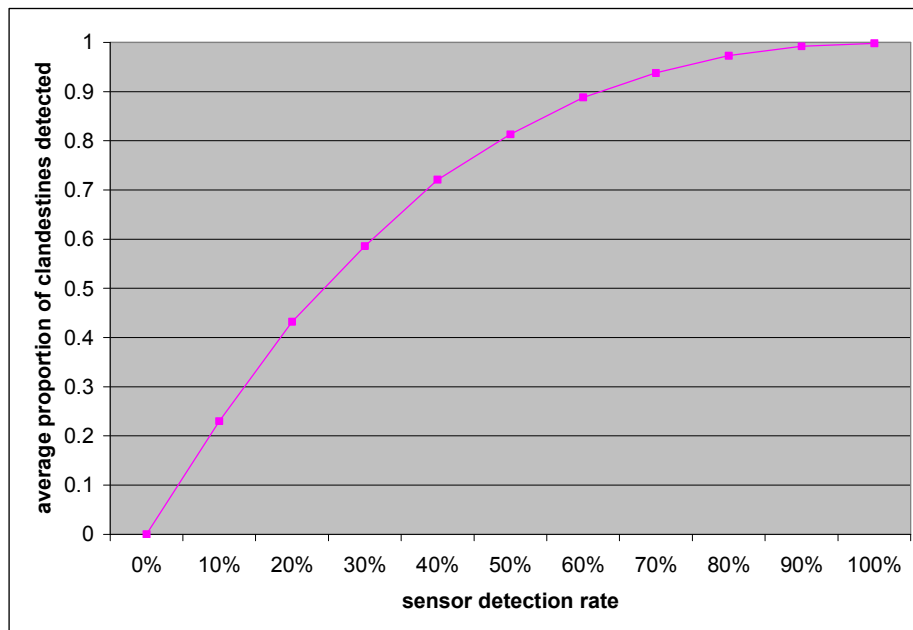


Figure 7: Results from the experiment: sensor detection rates vs. proportion of clandestines detected

average system performance but also want to gain an insight into its operations. We are currently working on the mechanisms to implement varying arrival rate and the consequences of these, i.e. congestion and varying service times. Once we have conducted our main data collection we will add some real data to it. Once this is done we will work on verification and validation of the simulation model.

ACKNOWLEDGMENTS

This project is supported by the EPSRC, grant number EP/G004234/1 and the UK Border Agency.

REFERENCES

- Brown, D.A., 1995. Human occupancy detection. Proceedings of the Institute of Electrical and Electronics Engineers 29th Annual 1995 International Carnahan Conference on Security Technology, 166-174. October 18-20, Sanderstead, Surrey, England.
- Closs, D.J., McGarrell, E.F., 2004. Enhancing security throughout the supply chain. Special Report to the IBM Center for the Business of Government.
- Dorndorf, U., Herbers, J., Panascia, E., Zimmermann, H.J., 2007. Ports o' call for O.R. problems. *OR/MS Today*. Available from http://www.accessmylibrary.com/coms2/summary_0286-30867578_ITM [accessed 01 April 2009].
- Epenshade, T.J., 1995. Using INS border apprehension data to measure the flow of undocumented migrants crossing the US-Mexico frontier. *International Migration Review* 29:545-565.
- Heidelberger, P., 1995. Fast simulation of rare events in queuing and reliability models. *ACM Transactions on Modeling and Computer Simulation* 5(1):43-85.
- Klock, B.A., 2005. Examination of possible technologies for the detection of human stowaways in air cargo containers. Final Report produced for Transportation Security Administration. Atlantic City: New Jersey, USA.
- Leone, K., Liu, R., 2005. The key design parameters of checked baggage screening systems in airports. *Journal of Air Transport Management* 11:69-78.
- Sekine, J., Campos-Náñez, E., Harrauld, J.R., Abeledo, H., 2006. A simulation-based approach to trade-off analysis of port security. Proceedings of the 38th Winter Simulation Conference, 521-528. December 03-06, Monterey, California, USA.
- Sky1, 2009. UK Border Force: Jargon Buster Part I. Available from: <http://sky1.sky.com/uk-border-force-jargon-buster-part-i> [accessed 01 April 2009].
- Tahar, R.M., Hussain, K., 2000. Simulation and analysis for the Kelang container terminal operations. *Logistics Information Management* 13(1):14-20.
- Turner, K., Williams, G., 2005. Modelling complexity in the automotive industry supply chain. *Journal of Manufacturing Technology Management* 16(4):447-458.
- UK Border Agency, 2009. Freight search figures for 2007/2008. Provided by the UK Border Agency.
- Wein, L.M., Liu, Y., Motskin, A., 2009. Analyzing the homeland security of the U.S.-Mexico Border. *Risk Analysis* 29(5):699-713.
- Weiss, W.E., 2008. Dynamic security: An agent-based model for airport defense. Proceedings of the 40th Winter Simulation Conference, 1320-1325. December 07-10, Miami, Florida, USA.
- Wilson, D.L., 2005. Use of modeling and simulation to support airport security. *IEEE Aerospace and Electronic Systems Magazine* 208:3-8.

XJ Technologies, 2009. XJ Technologies - Simulation Software and Services. Available from: <http://www.xjtek.com> [accessed 01 April 2009].

AUTHORS BIOGRAPHY

PEER-OLAF SIEBERS is a Research Fellow at The University of Nottingham, School of Computer Science. His main research interest is the application of computer simulation to study human-centric complex adaptive systems. This is a highly interdisciplinary research field, involving disciplines like social science, psychology, management science, operations research, economics and engineering. Furthermore, he is interested in nature inspired computing and agent-based robotics. For more information see <http://www.cs.nott.ac.uk/~pos/>

GALINA SHERMAN is a PhD student at City University, Cass Business School. Her current research is related to supply chain management, risk analysis and rare event modelling.

UWE AICKELIN is a Professor of Computer Science and an Advanced EPSRC Research Fellow at The University of Nottingham, School of Computer Science. His main research interests are mathematical modelling, agent-based simulation, heuristic optimisation and artificial immune systems. For more information see <http://www.cs.nott.ac.uk/~uxa/>

METRO SIMULATION AND OPTIMIZATION

A. Guasch^(a), D. Huguet^(b), J. Montero^(c), J. Figueras^(c)

^(a)LogiSim, Universitat Politècnica de Catalunya

^(b)Siemens, Mobility Division

^(c)LogiSim, Universitat Politècnica de Catalunya

^(d)LogiSim, Universitat Politècnica de Catalunya

^(a)toni.guasch@upc.edu, ^(b)david.huguet@siemens.com, ^(c)monty@fub.upc.edu, ^(c)jaume.figueras@upc.edu

ABSTRACT

Manless Train Operation (MTO) control systems are being introduced in metro lines around the world. In this paper we present an MTO metro simulation system developed for the analysis of metro logistic operations and an optimization model for obtaining Optimal Daily Circulation Plans (ODCP) which reduces the energy consumption by improving the synchronization between accelerating and breaking metros, with energy regenerative braking systems, within electrical substation areas. The target system for this research project is the line L9 of Barcelona metro network.

Keywords: metro simulation; metro optimization; manless train operations; regenerative braking

1. INTRODUCTION

With smaller inter-station distances, metro operation is essentially of start/stop nature. Due to frequent acceleration and braking requirements, energy demand is very high. The quantity of energy consumed by trains is influenced by a wide range of factors, which can be grouped as (i) Design of network, (ii) Design of trains & (iii) Service planning operation (Nielsen 2005, Joshi 2008). Hence, optimization of overall system design in order to control consumption of electricity becomes essential.

Traction accounts for about 60-80% of total energy consumption in a Metro system. In order to reduce Metro consumption, modern trains incorporate regenerative braking systems. Metro railways worldwide have reported an average of about 20% saving in traction energy on account of regeneration. It also helps to reduce heat load inside tunnel and thus reduce air conditioning load if available.

The first part of this paper focuses on the optimization of the service operations in order to maximize the energy recovered by the regenerative braking systems in a metro network. Figure 1 shows that, in theory, up to a 40% of the train input energy can be recovered by the braking system. However, in practice, this depends on the overall design of the system. The paper focuses in the situation where the regenerated energy can only be reused by trains moving in the same substation area. If the regenerative energy is not needed in the area, it is dissipated by the electrical

network. In these cases is important the synchronization between accelerations and brakings within a substation area. Thus, the energy flows from the braking train to the acceleration train. The target optimization problem presented here is obtaining an ODCP that maximizes the number of synchronized operations taking into account the service operational restrictions.

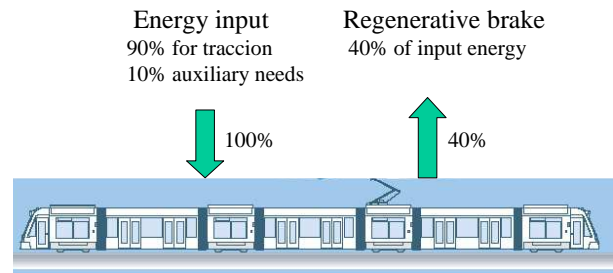


Figure 1: Regenerative brake energy balance

The capacity of a traditional metro system for following ODCM is quite limited due to human intervention in the system. Therefore, the results of this project are only effective in Driveless or Manless metro systems (Figure 2).

Unlike conventional or driverless metro systems, a manless system consists of fully automated trains, without any driver or attendant on board the vehicles. Over the past decades, several manless systems were designed and put into operation around the world, offering a high quality of service and generating increased revenue.





	START/STOP	DOORS	INCIDENTS
 ATP	driver	driver	driver
 ATP + ATO	automatic	driver	driver
 DTO (Driverless)	automatic	automatic	On-board assistant
 MTO (Manless)	automatic	automatic	automatic

Figure 2: Metro systems

The second part of this paper is devoted to description of a metro simulator which is being developed to analyze manless train operations:

- Evaluation of ODCM
- Analysis of specific operations, for example, shuttle metros for mass social events.
- Analysis of degraded mode operations. The degraded mode is active when normal operations cannot continue due to incidents, failures or accidents.

2. DESCRIPTION OF THE TARGET SYSTEM

Next figure shows the L9 Barcelona metro line. With its 46,6 km will be the first MTO line in Spain and, up to know, the largest MTO line in Europe. It has 51 stations equipped with automatic doors which prevent fatal accidents at the stations.

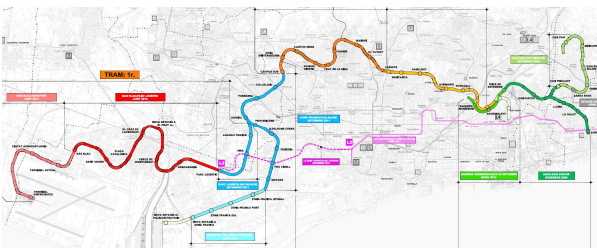


Figure 3: Barcelona Metro L9

A Daily Circulation Plan (DCP) is composed of a set of Line Services (LS). A LS specifies a set of consecutive station stops and the absolute starting time after each stop. Automatic Train Supervision (ATS) defines the movement time between stations in order to accomplish the LS. The movement can be chosen from a limited set of predefined times named:

- Slow time
- Normal time
- Rapid time
- Fast time

There is a difference of 5 seconds between each time level. LS are predefined using Normal Times, however, in case of a train disturbance; the Automatic Train Control (ATC) system can switch to a faster time to recover lost time. If the train is unable to recover the LS switching to a faster time, an order is issued from the ATS to all metros in order to resynchronize with the delayed one. The ability to follow specified LS is critical for minimizing energy consumption by synchronizing accelerating and breaking trains within the same electrical substation area. The MTO technology used in the L9 line guarantees that this will be true most of the time.

Since the L9 line is still under civil construction, the electrical network configuration is not known yet. Therefore, the electrical network substations shown in next figure composed by 6 electrical substations is only an assumption.

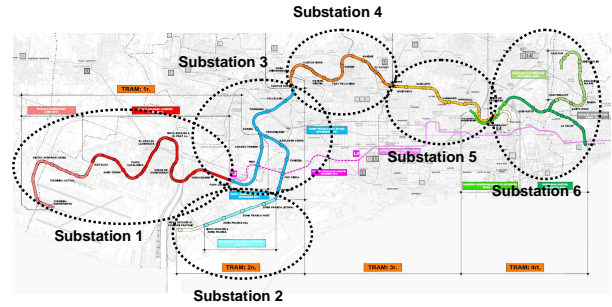


Figure 4: L9 electrical substations

The operational strategies have not been devised yet. We have assumed that there are three different metro lines shown in next figure. Line L9A that goes from northwest to southeast and vice versa; line L9B that goes from southwest to northeast and vice versa and; L9C that moves in the central section that covers central Barcelona.

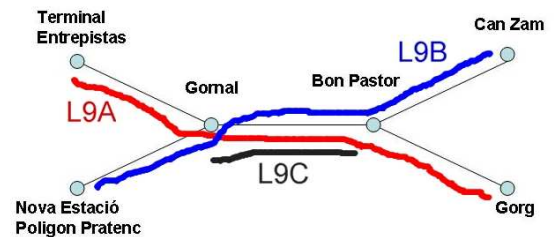


Figure 5: L9 lines

The above assumed configuration can be upgraded as soon as real configuration becomes available.

3. OPTIMIZATION MODEL

A first optimization model was developed using a discrete time model with a period of one second. The first results were unsuccessful due to the lack of convergence of the CONOPT solver.

The second approach is an event oriented model that uses data obtained from the simulation or real data if available. The data needed for every direct move between two stations is: minimum stop time at starting station; acceleration time (t_a), running time (t_r) and breaking time (t_d). The simplified model equations are:

$$\begin{aligned}
 t_{r,i} &= t_{r-1,i} + t_{s_{r,i}} & \forall r = r_s; i = 1, n \\
 t_{r,i} &= t_{r-1,i} + t_{a_{r,i}} & \forall r = r_a; i = 1, n \\
 t_{r,i} &= t_{r-1,i} + t_{r_{r,i}} & \forall r = r_r; i = 1, n \\
 t_{r,i} &= t_{r-1,i} + t_{d_{r,i}} & \forall r = r_d; i = 1, n
 \end{aligned} \tag{1}$$

Where t_s is the stop variable constrained by the minimum stop time and the desired maximum stop time. r is the set of events and r_s , r_a , r_r and r_d are the subset of the end of stop time, end of accelerating time, end of run time and end of breaking time respectively. i indexes the consecutive line services. In the first model configuration the optimization goal is to obtain

the set of stop times that maximizes the synchronization between acceleration and breaking periods.

The main problem constraints are:

$$\begin{aligned} t_{r,i} - t_{r-1,i} &\leq tp + \Delta tp & \forall r = r_s; i = 1, n \\ t_{r,i} - t_{r-1,i} &\leq tp - \Delta tp & \forall r = r_s; i = 1, n \\ ts_{r,i} &\in [10,20] & \forall r = r_s; i = 1, n \end{aligned} \quad (2)$$

The first two constraints limit the time period between two consecutive metros to tp seconds plus/minus Δtp seconds. For example, if the time period is 180 second we can constraint the time between metros to the [144,216] seconds interval. The third constraint states that the stop time at any station must be in the [10,20] seconds interval. The ability of the optimizer to obtain a good solution depends on the relaxation of the constraints. On the contrary, if we relax the constraints the level of service will be poorer.

Since the capacity of the solver is limited and the time it takes to obtain a solution is quite high (1 hour aprox.), we have limited the model time to approximately 2 and a half hours.

A set of constraints have been added to the model to obtain Partial Circulation Plans (PCP) or Optimal Partial Circulation Plans (OPCP) of an approximate duration of 30 minutes that have the quality that the position of the metros at the beginning of the period is equal to the position of the metros at the end of the period. Thus, if a metro is in the Camp Nou station at the beginning of the period, the same or another metro will be at this position at the end of the period. If a PCP or OPCP satisfies this constraints a larger PCP or OPCP can be obtained by replicating the plan as many times as needed.

The size of the optimization model is between 30.000,0 and 45.000,0 equations depending on the working configuration.

4. OPTIMIZATION RESULTS

Next figure shows an OSCP. Metros going to the suburbs have a time period of 8 minutes. In the central sector, the period between metros is 2 minutes. Thus, L9A, L9B and L9C metros merging in the central sector are separated by the period of 2 minutes plus or minus the tolerance accepted in the constraint equations.

Table 1 shows a comparative of the results obtained in this case. The columns are, from left to right,

1. In the first two rows we consider that the movement time between stations is constant. In the next rows we want to see the impact of having a variable time between stations which is constrained by the Δtr value specified in the first column. In this case, the modified equations are,

$$\begin{aligned} t_{r,i} &= t_{r-1,i} + tr_{r,i} + \Delta tr_{r,i} & \forall r = r_s; i = 1, n \\ \Delta tr_{r,i} &\in [-\Delta tr, \Delta tr] \end{aligned} \quad (3)$$

This is a simplified equation since the time variation with respect to the Normal Time is only attributed to the running time. The capability of playing with the running times eases the task of synchronizing accelerating and breaking metros.

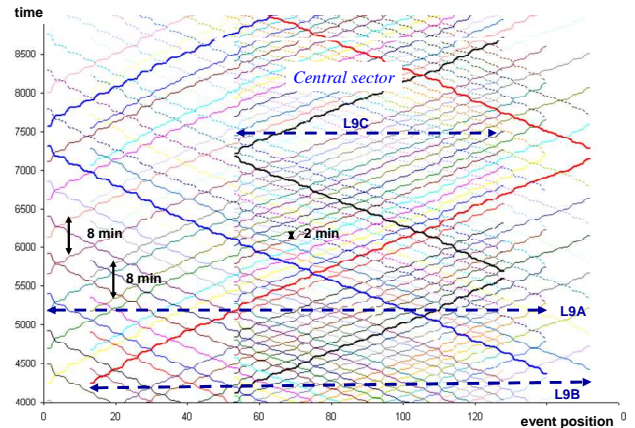


Figure 6: OSCP

2. The second, third and fourth columns show the specified time period for each metro line. The three following columns show Δtp period tolerance with respect to the specified time period.
3. Next column shows the total breaking time (tbt). tbt is the accumulated breaking time for all metros during one hour approximately.
4. Column number 9 shows the total non synchronized breaking time. The optimization function objective is the minimization of this value.

Table 1: Comparative results (seconds)

Δtr	tp (Central sector)	tp (L9A)	tp (L9B)	Δtp (L9C)	Δtp (Central sector)	Δtp (L9B)	tbt	nsbt	Improvement	% improvement
	120	480	480	20	60	60	32160	12236		
0	120	480	480	20	60	60	32099	10005	2231	6.9
2.5	120	480	480	20	60	60	32139	7228	5008	15.5
5	120	480	480	20	60	60	32180	5855	6381	19.8
7.5	120	480	480	20	60	60	32159	5304	6932	21.5
	180	480	360	27	180	54	28940	12670		
0	180	480	360	27	180	54	28639	10364	2306	8
2.5	180	480	360	27	180	54	28619	8898	3772	13.1
5	180	480	360	27	180	54	28600	8370	4300	15
7.5	180	480	360	27	180	54	28580	8759	3911	13.6

5. Column number 10 shows the improvement in synchronization as a result of the optimization.
6. The last column shows the percentage reduction of the total non synchronized breaking time with respect to the total breaking time.

The first block of results has a central sector time period of 2 minutes while the second block as a central sector period of 3 minutes and the L9B time period has been reduced to 6 minutes instead of 8 minutes in the first block.

The first experiment of each block shows the results without optimization. The only requirement of the solution is satisfying the equations and the constraints. The second experiment of each block increases the synchronization by optimizing the stop times at each metro station by minimizing the non synchronized breaking time. Approximately an additional 36 breaking minutes are synchronized with accelerating metros in one hour period. This is a between 6.9% and 8% improvement.

The three following experiments of each block suppose that the running times can be adjusted with a limit of +/-2.5, +/-5 and +/-7.5 seconds between each station. In the last experiment of the first block 115 breaking minutes are synchronized with accelerating metros in one hour period. This is a between 21.56% improvement.

The results of the last experiment of the second block should be improved since the solution obtained for the solver is not good enough. The main difficulty for obtaining a quasi optimal solution is that the cost function is non convex. We have performed a sensitivity analysis giving the solver different initial values for the same model. In general, the solver performs reasonably well in all cases. Next figure shows the value of the cost function given different starting points in the stop times at metro stations direction west and east.

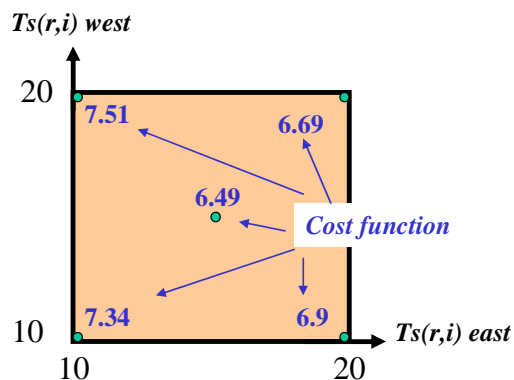


Figure 7: Sensitivity analysis

Significant synchronization improvements are obtained when the movement times between metro stations can be adjusted within a certain time interval. Unfortunately, L9 ATC only allows four different times: slow, normal, rapid and fast. Since the fast mode is left for delay correction, we only can use three different times instead of a continuous interval. We plan, not done yet, to use the simulator to measure the final improvement after discretization of the continuous interval into three possible values.

We also plan to try the CPLEX solver to solve the problem. This will allow coping with slow, normal and rapid discrete movement times directly. However, we are afraid of the capability of the solver to handle the non convex cost function.

5. SIMULATION

A manless metro simulator has been developed in the project for the evaluation of the metro logistic operations and for validation of the previous optimization results.

The basic continuous equations used to model the metro dynamics are,

$$T - R = m \cdot a \tag{4}$$

$$R = A + B \cdot v + C \cdot v^2$$

Where T is the traction effort shown in figure 8 and R is the resistance that the train has on its movement. The coefficients A, B and C depend on the mechanical resistance, aerodynamic drag and grade resistance. Using these equations, the metro acceleration in traction mode is,

$$a = \frac{1}{m} [T - (A + B \cdot v + C \cdot v^2)] \tag{5}$$

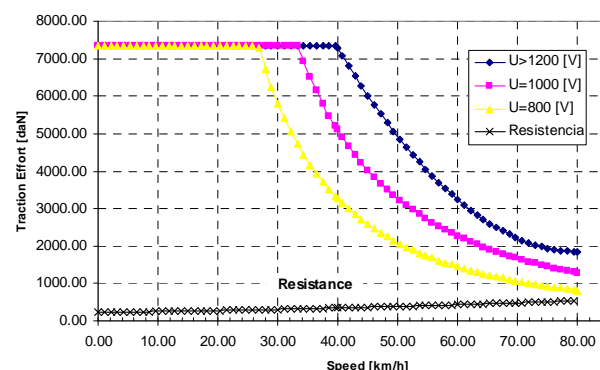


Figure 8: Traction effort

Barcelona L9 metro network uses the moving block control system. Under a moving block system, computers calculate a 'safe zone' around each moving train that no other train is allowed to enter. The system depends on knowledge of the precise location and speed and direction of each train, which is determined by a combination of several sensors: active and passive markers along the track and trainborne tachometers and speedometers. With a moving block, lineside signals are unnecessary, and instructions are passed directly to the trains. This has the advantage of increasing track capacity by allowing trains to run closer together while maintaining the required safety margins.

Figure 9 shows the dynamics of two metros moving very close. As metro1 accelerates, the distance (x1-x2) between both metros increases. When metro1 stops at the first station and metro2 moving behind

accelerates, the distance is reduced. Since metro2 does not stop at the any station, it catches metro1 at the second station. At that point the distance between metros is the length of the metro1 plus a security margin. Metro2 starts moving as soon as metro1 starts but the security distance has to increase to avoid crashing in case of sudden stop of metro1.

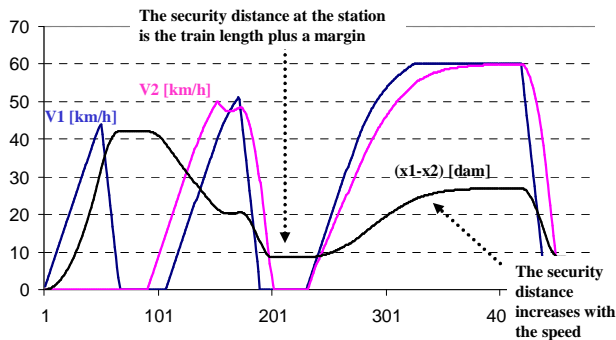


Figure 9: Moving block dynamics

6. SIMULATION RESULTS

Up to know, the simulator has been used,

- to obtain the date needed for the optimization
- to check that the OPCP generated by the optimizer can be followed by the simulated metro network.

Though the simulator is already operative, further work is needed along the next months in order to add further experimentation capabilities,

- The first new set of experiments will measure the synchronization time taking into account that the movement time between stations has three discrete candidates.
- The second experiment set will analyzed and propose working strategies in degraded mode. For example, a probable case is the blocking of a metro in a station. An alternative operational mode has to be proposed to avoid paralyzing the metro network. One possibility shown in next figure is to let metros in both directions to share the same track. A second possibility is to use the first arriving metro as a shuttle using the single track zone in exclusivity.

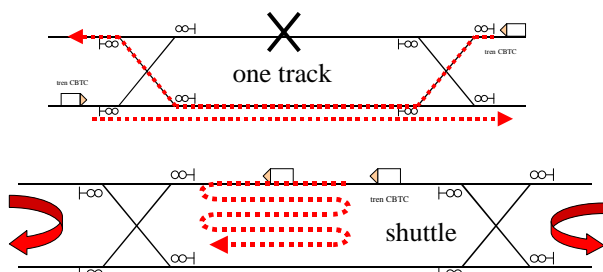


Figure 10: Single track operation in degraded mode

- The third experiment set will propose new operational strategies to cope with specific needs such us crowded events. Two specific events will attract our attention. The mobility needs at the exit

of the Camp Nou soccer stadium and the mobility needs of the Barcelona Fair. The Camp Nou soccer stadium has a capacity of 98 thousand people. Soccer games are usually played on Saturday or Sunday evening. Since L9 will have spare capacity in this time zone, we will try to insert shuttle trains that have the first stop at the soccer stadium and the next stops will be at specific metro stations. For example, stations with the possibility of changing lines. The needs for the Barcelona Fair are more related to having direct metros to the airport at specific times.

7. CONCLUSIONS

A metro optimization model and methodology has been developed with the goal of increasing the synchronization between acceleration and breaking metros in manless metro networks with energy regenerative breaking. Further work is needed to estimate the potential saving in electrical terms. However, we think that event small savings will payoff in midterm since the execution of optimal circulation plans does not require any investment by the metro operators.

A metro simulator has been developed with the objective of validating the optimal circulation plans and designing operational procedures for handling degraded mode or special scenarios.

ACKNOWLEDGMENTS

This work is being funded by the Departament d'Innovació Universitat i Empresa (Generalitat de Catalunya, Spain) and the Ministerio de Industria, Turismo y Comercio (Spain).

We wish to thank the master students Joan Antoni Navarro and Alvaro Sanchez-Muro for their contribution to the project.

REFERENCES

- Joshi S.S, Pande OH, Kumar A., 2008. *Regenerative Breaking in metrol Rolling stock*. International Seminar on Emerging Technologies & Strategies for Energy Management in Railways. New Delhi, 2008.
- Nielsen J. B., van Essen H.P., den Boer L.C., 2005. *Tracks for saving energy. Energy saving options for NS Reizigers*. Delft, CE, July 2005

AUTHORS BIOGRAPHY

Born in 1958, Dr. Antoni Guasch is a research engineer focusing on modelling, simulation and optimization of dynamic systems, especially continuous and discrete-event simulation of industrial processes. He received his Ph.D. from the UPC in 1987. After a postdoctoral period at the State University of California (USA), he

becomes a Professor of the UPC (www.upc.edu). He is now Professor in the department of "Ingeniería de Sistemas, Automática e Informática Industrial" in the UPC. Since 1990, Prof Guasch has lead 35 industrial projects related with modelling, simulation and optimization of nuclear, textile, transportation, car manufacturing and steel industrial processes. Prof. Guasch has also been the Scientific Co-ordinator and researcher in 7 scientific projects. He has also been involved in the organization of local and international simulation conferences. For example, he was the Conference Chairman of the European Simulation Multiconference that was held in Barcelona in 1994. Prof. Guasch has recently published a modelling and simulation book that is being used for teaching in many Spanish universities. Prof. Guasch currently research projected sponsored by Siemens is related to the development of power management optimization algorithms for the Barcelona new L9 subway network. L9 will be the first metro line without driver of Spain and one of the largest (45 km) and modern automatic line without driver of the World. Prof. Guasch is also contributing to the development of toopath (www.toopath.com), a web server system for the free tracking of mobile devices.

Born in 1975, Dr. David Huguet works in Mobility division of SIEMENS as a Project Research Manager. Dr. Huguet's work is focus on simulation and modelling of metro and tramway for optimizing their kinematic profile and for reducing the traction energy. He is actually professor of the Mechanical Department of the Polytechnic University of Catalonia, where he received his Ph.D. in 2005 for his work on Fluid Mechanics. Actually Dr. Huguet is involved in SIMUL9 and TRAM projects in collaboration with the UPC researchers.

Jordi Montero, born in 1968, had his degree in Computer Science in 1998, has been PAS of UPC since 2000. His research interest includes discrete simulation system and he has working in several projects for different enterprises -Aena, Agbar, Almirall Prodesfarma, Damm, Indra, Siemens,...- and sectors: clinical trail, airport, transports, manufacturing, pharmaceutical.

Jaume Figueras, born in 1974, had his degree in Computer Science in 1998. His research is in Automatic Control and Computer Simulation an Optimization. He has designed and developed CORAL, an optimal control system for sewer networks, applied at Barcelona (Spain); PLIO, an optimal control system and planner for drinking water production and distribution, applied at Santiago de Chile (Chile) and Murcia (Spain). Nowadays He participates in different industrial projects, like the power consumption optimization of tramway lines in Barcelona with TRAM and SIEMENS and the development of tooPath (<http://www.toopath.com>) a free web tracking system of mobile devices. He is also the local representative of

OSM (<http://www.openstreetmap.org>) in Catalonia and participates in different FOSS projects.

MODELING AND SIMULATION OF AN ULTRA-STABLE CRYOSTAT FOR NAHUAL

D. Sosa-Cabrera^(a) and F.J. Fuentes^(a)

^(a)Instituto de Astrofísica de Canarias (IAC). Vía Láctea s/n. 38200 La Laguna, Tenerife, Spain.

dario@iac.es and [ffj@iac.es](mailto:fjf@iac.es)

ABSTRACT

NAHUAL is a cryogenic high resolution near infrared echelle spectrograph for the *GTC* (Gran Telescopio de Canarias). The main goal of the instrument is to deliver high-precision radial velocities for cool stars at IR wavelengths. *NAHUAL* is conceived as a highly specialized instrument, capable of finding habitable planets around red dwarfs (Martín 2008).

IR instruments must operate under ultra-cold temperatures in temperature-controlled vacuum containers called dewars or cryostats to reduce disruption by heat noise.

The use of LN2 instead of mechanical coolers to cool the instrument is mandatory to avoid noise vibrations. *NAHUAL* concept is adapted from *GIANO@TNG*, with a requirement of thermal stability in the optical bench a factor 10 above *GIANO*'s one: ~ 0.1 K.

The modeling and simulations carried on at *NAHUAL*'s conceptual design are presented in this paper.

Keywords: telescope instrumentation, Finite Element Analysis, cryostat, thermal stability

1. INTRODUCTION

NAHUAL is a cryogenic high resolution near infrared echelle spectrograph for the *GTC* (Álvarez et al. 2006). This paper describes the considerations taken into account for calculating *NAHUAL*'s optical elements displacements during observation and aims to determine the thermal stability required at the optical bench. At the same time the model will be fed up with the *Zemax* (a software that optical engineers and designers for lens design and other applications) results in order to define structurally the model.

A FEA model has been developed in order to estimate, as accurately as possible, the thermal budget and structural deformations at *NAHUAL*'s Cold Structure, and at the same time to validate the accuracy of the model with respect to the accuracy demanded by the specifications of the project. After the thermal budget is calculated, simulations have been down in order to estimate the deformations suffered by the different optical elements of *NAHUAL*'s Optical Bench,

and to obtain a thermal stability restriction according to the image pixel stability specification of the project.

NAHUAL specifications include the observation of bright point sources ($J < 12$) at very high spectral resolution ($R > 50,000$) with very high image quality and spectral stability (better than 1/1000 pixels) enabling high accuracy RV measurements. Having in mind a thermal stability of mili-kelvin during one night of observation as starting point, we will try to make the translation of the specifications to thermal stability.

One of the goals of these simulations is to state whether it is needed or not to fabricate a prototype for the instrument.

The analyses have been computed mostly in an Intel Pentium IV, 3 GHz; the computational times referred to along this document are related to that hardware.

2. THE MODEL

The model has been carried out using the *ANSYS 10.0* software of Finite Element Analysis (ANSYS 2005). A steady and a transient thermal state will be studied. The whole procedure to analyze the thermal steady state consisted of analyzing the incoming heat loads both by thermal conduction of the parts of the model and by radiation. We will study the model in order to investigate the image stability. During the observation period, the consumption of the LN2 will be analyzed. The refilling of the deposit is not an option as there is enough volume for the LN2 vessel to last for more than one night of observation, and it would change the equilibrium conditions at the vessel changing the temperatures at the optical bench.

Before developing the final model presented in this study, we compared the possibilities of having a double LN2 vessel, as in *GIANO* instrument (Mochi et al. 2008), or a single one, and the latter option was adopted. FRIDA thermal model has also been a reference for this work (López 2008).

The Finite Element Model for *NAHUAL* was built considering the following constituent elements:

- One LN2 vessel modeled as a box. At the base the trusses and cold shield are attached. The upper surface of the vessel represents the cold bench, where the optical instruments lie.

- The outermost cylindrical body (here named cryostat).
- A cold shield attached to the LN2 vessel. This shield will be covered with Mylar, however this effect will be modeled adapting the emissivities of the surfaces.
- A floating shield placed in between the cryostat and the LN2 vessel-cold shield.
- The trusses were simulated applying heat to some nodes of the model.

In figure 1, a cross-section of the model is plotted. The dimensions are summarized in Table 1.

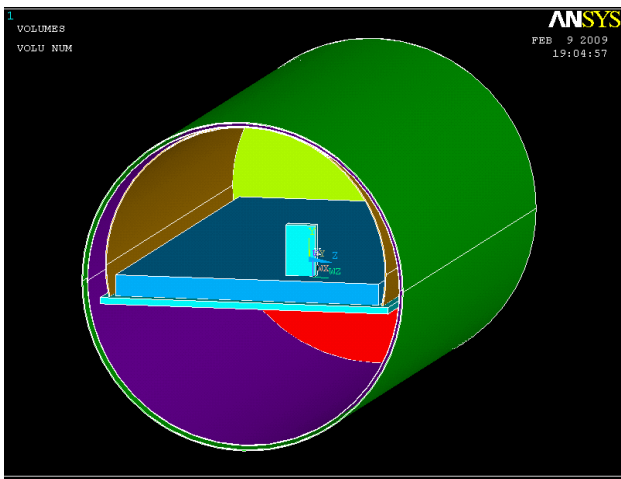


Figure 1: Cryostat Model

Table 1: Model Dimensions

Component	Height - Radius	Width	Length	Thick-ness
Cryostat body	851	1702	2820	4
Floating shield	827	1654	2571	2
Cold shield	673	1322	2518	5
LN2 vessel	80	1230	2426	20
Base	30	1390	2586	-

3. THERMAL SIMULATIONS

Thermal conductivities for the materials involved in the construction of the cryostat were defined in the model as a function of the range of temperatures of interest.

For the analysis, definitions of thermal conductivities were also required and they were all defined based on those reference temperatures.

The thermal budget has been calculated taking into account the following conditions:

1. The incoming heat loads by thermal conduction through trusses (the support bars) while in order to simplify the model, wiring and pipes are not considered. Trusses effect was simulated as hiperstatic and isostatic boundary conditions. The results we present next consider the former approach as the incoming heat is

derived directly to the deposit (trusses would be placed at the bottom of the base, and not at the sides).

2. The radiation loads taking place through:

- the cryostat,
- one adiabatic shield,
- one cold shield and
- the LN2 vessel
- the window has not been modeled at this stage of the design and heat has been applied to some node at the first optical element

3. Radiation can be calculated considering different surface-finishing for the components mentioned before. At this stage one possibility is considered for each surface. We have studied two cases: one with the theoretical emissivities and other with a set of conservative ones:

Theoretical

- $emis_CS = eAl_My77 = 0.0036$ Cold Shield
- $emis_FS = eAl_ep293 = 0.075$ Floating Shield, and Vessel
- $emis_Cryo = eAl_ep293 = 0.075$ Cryostat

Conservative

- $emis_CS = 0.1$ Cold Shield
- $emis_FS = 0.1$ Floating Shield, and vessel
- $emis_Cryo = 0.25$ Cryostat

The model is built with 3 enclosures. An enclosure is a set of surfaces radiating to each other:

- Enclosure 1: external Vessel and inner Cold Shield surfaces.
- Enclosure 2: external Cold Shield, inner Floating Shield and bottom surface of the Base.
- Enclosure 3: external Floating Shield and inner Cryostat.

ANSYS uses the definition of an enclosure to calculate view factors amongst surfaces belonging to an enclosure.

4. The lowest temperature in the cryostat is supposed to be the LN2 phase change temperature. This temperature depends on the pressure at the vessel, and how to measure or estimate that pressure is a critical issue of the design. According to table 6.22 from Electro-Optical Components, (Rogatto 1993), this temperature is 77.37 K at 1 at. At this moment it has been assumed to be 77K and is applied at the inner part of the vessels for the steady analysis.

5. It must be underlined that the floating shield is not attached in the model to other parts so no conduction heat is included for this element at this point. This assumption was done for the sake of model simplification and taking into account that the most

significant source of heat exchange through this component is due to radiation from the cryostat ambient temperature and not due to conduction through the bars.

3.1. Meshing

Different meshes have been tested. In some cases some of the volumes were added and in some occasions the nodes were merged or coupled. In figures 2 to , some of the meshes used for the base and the vessel are shown.

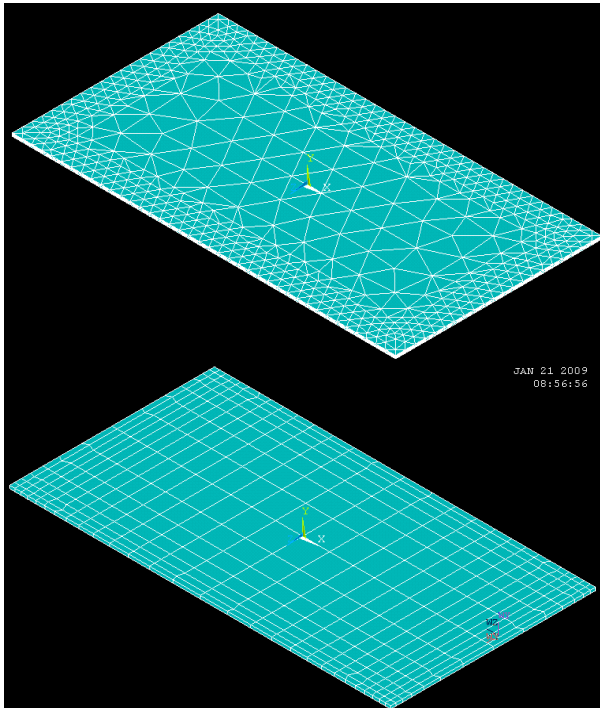


Figure 2: Tetrahedral and Hexahedral meshing for the base

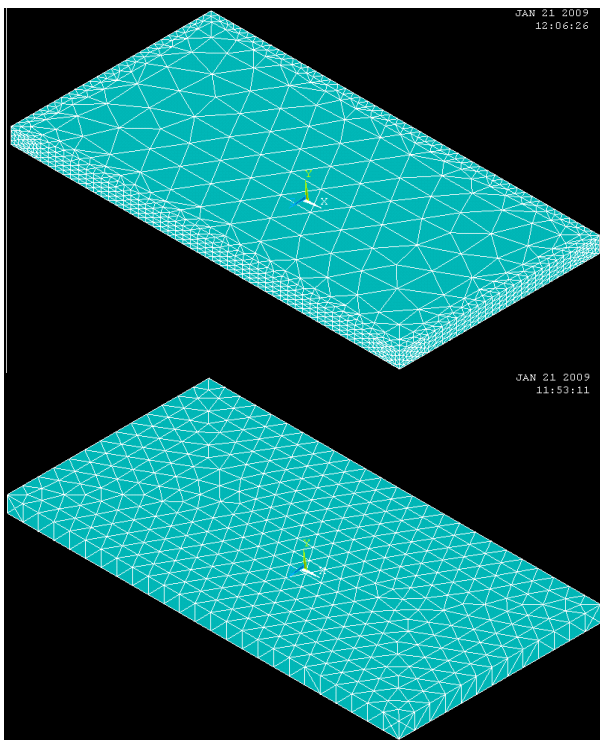


Figure 3: A coarse refined mesh and a finer homogeneous mesh for the vessel

3.2. Static Analysis

The steady state thermal model was carried out taking into account two boundary temperatures to determine the heat budget once the equilibrium of the whole cryostat is reached. The thermal conditions stated are the LN2 tank temperature of 77 K and the outermost environmental temperature of 281.5 K.

In figures 2 and 3, the thermal distribution of temperatures is plotted at the optical bench, the upper part of the LN2 vessel. The maximum temperature for both cases is 77.004 K, but reaching a wider area for the case where the LN2 deposit has been simulated empty, i.e. the 77 K BC is not given to all the nodes of the deposit but only those at its bottom part.

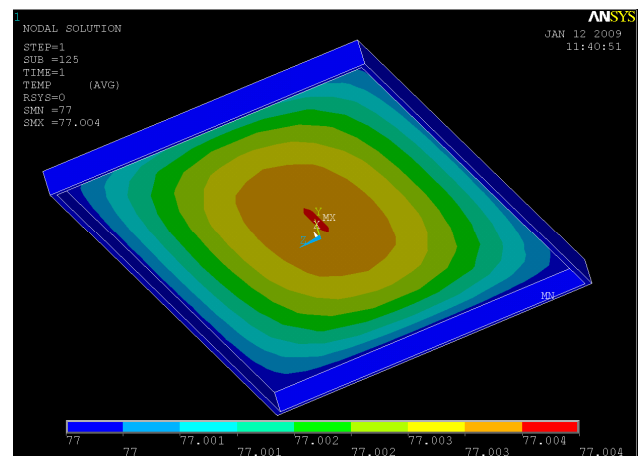


Figure 4: Optical Bench – Full Deposit

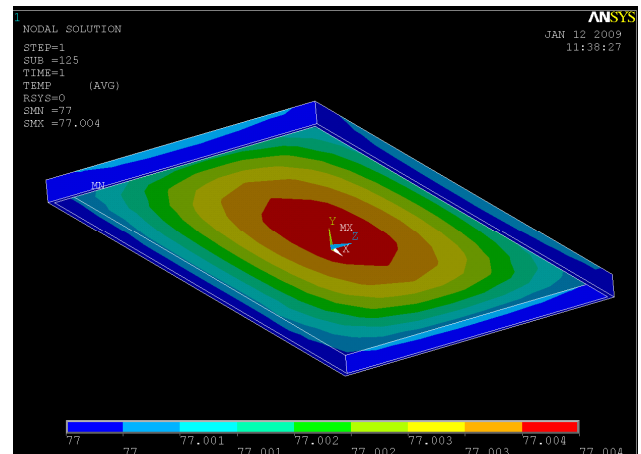


Figure 5: Optical Bench – Empty Deposit

The previous plots are based on a simulation with the theoretical emissivities while the next one, has been calculated with the conservative set. The real results are thought to be between these two situations. The maximum temperature, at the center of the optical bench rises 77.022 K.

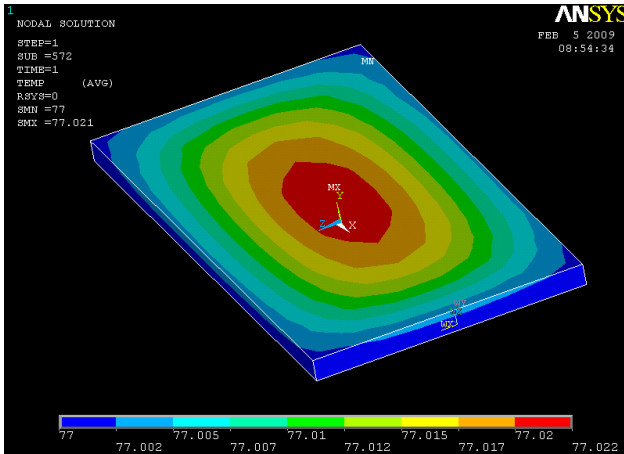


Figure 6: Optical Bench – Empty Deposit – Conservative Emissivities

3.3. Transient Analysis

For the transient study, a set of temperatures according to GTC data is considered. The maximum variation is 2.4 K/ 2 hr or the minimum and maximum temperatures are used (292 – 271 K).

A priori the decision made about the LN2 vessel is not to refill it during observation because that would change the equilibrium and would be worst for the thermal stability. In this model the evaporation has not been taken into account. Instead we have compared different situations in which the vessel is full or empty through static analysis.

Below, the curve used as BC for the environmental temperature affecting the cryostat is plotted. The curve is determined from the maximum and minimum temperatures, as it results in a worst scenario than the derivative restriction. It represents 24 hours (time in minutes at the graph) starting with the night period, indeed which is important for us.

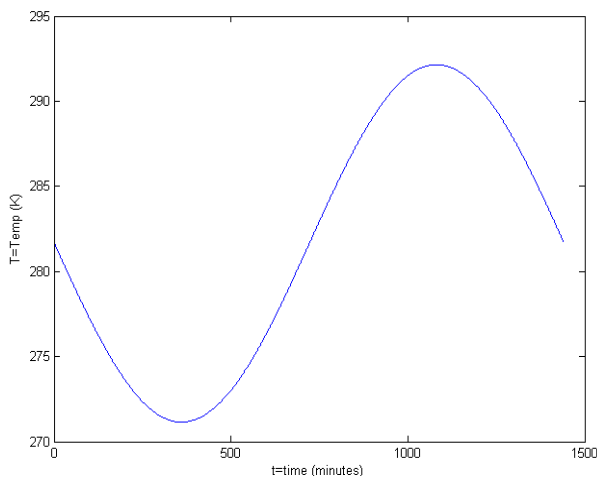


Figure 7: Curve: $T=281.65-10.5*\sin(0.00436*t)$

Looking at figures 5 and 6, the temperature curves resulting from the transient analysis with conservative emissivities can be compared.

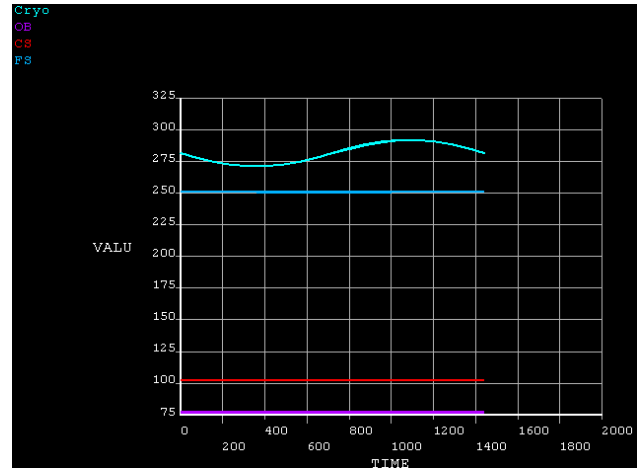


Figure 8: Temperatures at different nodes of: the Cryostat, the Floating Shield, the Cold Shield and the Optical Bench (from top to bottom)

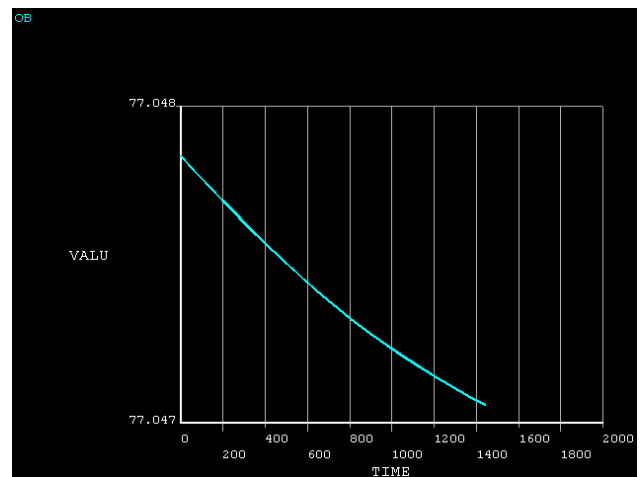


Figure 9: Temperature at a node centered on the optical bench

4. STRUCTURAL SIMULATIONS

After the thermal analysis, the next step is to estimate the displacements experimented by the model due to the thermal loads, the weight of the elements and the pressure inside the LN2 vessel (outside the Vessel is considered 0). We have solved this analysis only for the set of Base-Vessel-CS. The rest of the elements, i.e. the FS and the Cryostat have been “killed” (ANSYS command) for this analysis as we are interested on the displacements of the Cold Base in order to translate them to optical stability of the light.

We have modeled isostatic restrictions on the lower Base, as if isostatic trusses were supporting the Cold Mass. Besides, it was needed to rerun the thermal analysis adding the Base and the Vessel, therefore being one volume. Visualizing the results is then less convenient but it is necessary due to unconvergence problems that occur when the displacements affect both volumes making them overlap or cross each other.

The results presented below, are just an example to understand the capabilities and to prepare the pipeline

for iteration. We have not taken into account the weight of the LN2 and the radiation of the window. Later on, we will refine the model.

We have considered different pressures inside the Vessel in order to define the limit of the pressure variation to control the deformations and to meet our specifications. One important part of the project is to define the control system to meet the specifications.

In the following plots we present some images with the vertical displacement U_y of the Vessel nodes. The most important for us are the displacements of the Optical Bench, where the optics rest. ANSYS allows plotting into a text file the displacements at each node; U_x , U_y and U_z and also the x , y , z coordinates of those nodes. Two *MatLab* scripts, one reading the coordinates and numbers of the nodes and the other the displacements, are also used. These codes convert the *.lis* file into a set of arrays containing the different variables.

The results show a maximum vertical displacement 425 μm for 101 kPa and 459 μm for 109 kPa. We estimate a maximum of 101.96 kPa the maximum pressure, to meet the specifications. The maximum displacements occur in the middle of the Cold Bench. As said before, the final model will be reinforced structurally and these results are still at the conceptual stage.

To work with *Zemax* we will give the results for the full and the empty LN2 deposit cases; compared with different pressures. The difference between both displacements will be the relative displacement during one night of observation. If we compare the full deposit case with the node positions before the cooling, we will have information on how to design the pieces in order to be aligned when cooled down.

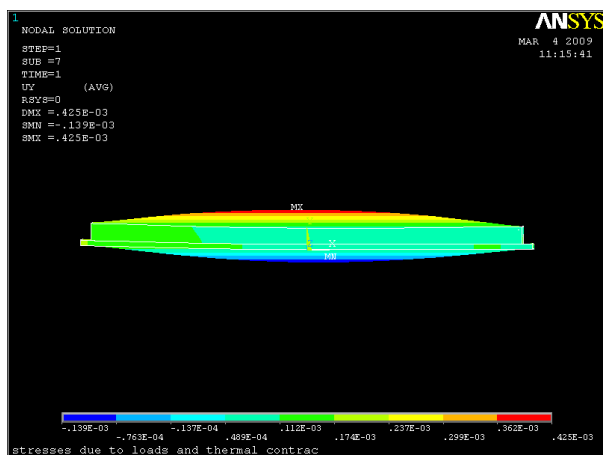


Figure 10: U_y displacements for an inner pressure of 101325 Pa

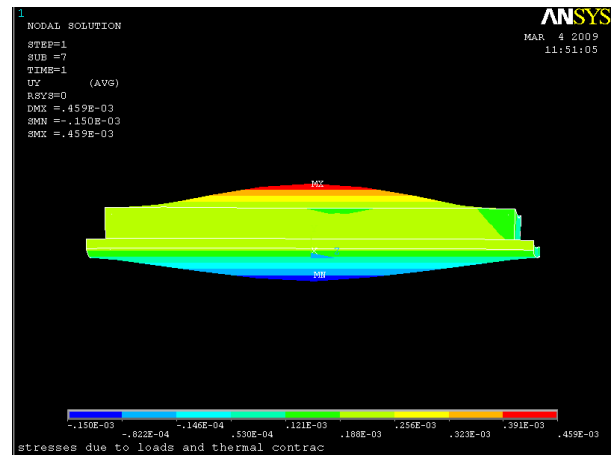


Figure 11: U_y displacements for an inner pressure of 109360 Pa

To obtain the deformations of each optical element, 3 nodes of the base are taken. At this first stage, as the elements are not modeled yet, the nodes are taken at the Optical Bench. The average displacement is calculated and the tilts from the normal vectors of the surfaces formed with the three nodes. With the tilts a transformation is made to estimate the displacements at the height of the optical path. Once *Zemax* is run we can infer whether the displacement suffered by different elements compensate or not, having the global optical specification. Finally the process iterates until meeting the stability requirements.

5. CONCLUSIONS AND FUTURE LINES

The FEA model has been developed in order to guess, as accurately as possible, the thermal budget of *NAHUAL's* Cold Structure, and to validate the accuracy of the model with respect to the accuracy demanded by the specifications of the project.

- The results' error is in the order of at least centi-kelvin, so, the model is not useful to predict thermal stability to the mili-kelvins. Even more when theoretical thermal properties assigned to the model are different to that obtained after fabrication and that the state of the art of the temperature sensors do not reach that precision.
- The optical bench is not affected by the changes in the external temperature; these variations are absorbed by the floating and the cold shields except in the worst conditions modeled where it is stable to the centi-kelvin in 24 hours.
- The meshing does not affect the result at the optical bench in terms of maximum temperature.
- The difference between the 2 situations where the LN2 deposit is filled or almost empty does not change the maximum temperature at the optical bench, only its distribution. In the case

where the vessel is almost empty, there is a wider area of the optical bench at the maximum temperature.

At this moment it is under study the viability to fabricate a scaled prototype. The idea is to have a test bank and to develop a FE model that can be tuned with the prototype. A change made to the prototype would be updated in the model and new predictions could be made.

ACKNOWLEDGMENTS

We would like to acknowledge Carlo Baffa and his team at Osservatorio di Arcetri, and Jorge Lima and colleagues from University of Lisbon. Support for this project has been provided by the Spanish Ministry of Science via project AYA2007-67458.

REFERENCES

Álvarez, P., Castro López-Tarruella, J., and Rodríguez-Espinosa J.M., 2006. The GTC Project. Preparing the First Light. *Proceedings of SPIE Vol. 6267*, 626708

ANSYS, Inc. Theory Release 10.0, 2005

López, A., 2008. FRIDA: Mechanical Design and Manufacture COLIBRÍ. *UNAM Internal Document Code: FR/DD-MC/045*

Martín, E., 2008. NAHUAL: The GTC Habitable Planet Finder. *IAC Internal Document Code: NAH-P-0001*

Mochi, I., Oliva, E., Origlia, L., Baffa, C., Biliotti, V., Falcini, G., Giani, E., Gonzalez, M., Rossetti, E., Sozzi, M., Liffredo, M., Roveta G., and Roccaie L., 2008. Performances of the cryogenic system of GIANO-TNG, *Proceedings of SPIE Vol. 7014* 70143U-1

Rogatto, W.D. (ed.), 1993. The Infrared & Electro-Optical Systems Handbook, vol. 3, 204, copublished by Infrared Information Analysis Center & SPIE Optical Engineering Press.

AUTHORS BIOGRAPHY

Darío Sosa received his BS Degree in 2001 from the Industrial Engineer School, the University of Las Palmas de Gran Canaria (ULPGC). He has studied at the Institut Français de Mécanique Avancée (IFMA) for his Degree Project in collaboration with the Medical Engineering Area of the Instituto Tecnológico de Canarias (ITC). After working as an engineer in Riga, Latvia, he has coursed a Management of Technology Master at the University of Texas at San Antonio (UTSA) and stayed at the Department of Radiology of the University of Texas Houston (UTH) doing a

research for his thesis during 2004 and 2005. He has been linked to different research projects funded by the Spanish Government (SINERGIA Surgical Simulator and USIMAG-ICTs in Ultrasonography) and other EU projects (SIMILAR Network of Excellence). In 2008 he got his PhD in IT from the ULPGC. His thesis focuses on US medical imaging, and in particular, on Elastography. Since then, he is working as a mechanical engineer at Instituto de Astrofísica de Canarias, IAC. He is mainly working with NAHUAL Planet Finder for GTC and HARMONI for E-ELT instruments.

Javier Fuentes obtained his BS Degree in 1982 from the Aeronautics Engineering School, Politecnico University, Madrid. He is working since 1985 at the IAC Mechanical Engineering Group, at present having a Senior Position. Experience in the design of astronomical instruments, includes the high resolution lab at the WHT, a 32 elements IR Camera, a Solar Correlation Tracker, CAIN, INTEGRAL, LIRIS, OSIRIS and EMIR. He has participated in 7 feasibility studies, including the IR camera for NOT, INTEGRAL, LIRIS, ABEL, CAIN-I, ATLANTIS and EMIR. He has been responsible of the mechanical team that developed CAIN, INTEGRAL and LIRIS. He is currently the Systems Engineer of FRIDA and NAHUAL and he is working too with HARMONI.

OPTIMIZATION APPROACH TO POWER LIMITS IN FLOW ENERGY SYSTEMS

Stanislaw Sieniutycz

Faculty of Chemical Engineering, Warsaw University of Technology
00-645 Warsaw, 1 Warynskiego Street, Poland
sieniutycz@ichip.pw.edu.pl

ABSTRACT

We develop a thermodynamic approach to modeling and power optimization of nonlinear energy converters, such like thermal, solar, chemical engines, and fuel cells. Thermodynamic principles lead to converter's efficiency and limiting power. Efficiency equations help to solve problems of upgrading and downgrading of resources. While optimization of steady systems requires using of differential calculus and Lagrange multipliers, any dynamic optimization involves variational calculus and dynamic programming. The primary result of the static optimization is a limiting value of power, whereas that of the dynamic optimization is a finite-rate counterpart of the classical reversible work potential (exergy). In reacting systems chemical affinity constitutes a prevailing component of an overall efficiency, and, therefore, flux balances are applied to derive power yield in terms of an active part of chemical affinity. We point out the similarity of power limits in flow thermal machines and fuel cells.

Keywords: power limits, optimization, entropy, thermal machines, chemical engines, fuel cells

1. INTRODUCTION: POWER YIELD DRIVEN BY FLOW OF ENERGY AND RESOURCES

Transfer phenomena influence significantly performance of power yield systems and transformation of resources. In a process of power production shown in Fig. 1 two moving media interact through an energy generator (engine) and the process is propelled by diffusive and/or convective fluxes of heat and mass transferred through 'conductances' or boundary layers in the resource fluid ('upper' fluid 1) and in, say, an environment fluid ('lower' fluid, 2). In principle, both transfer mechanisms and values of conductances of boundary layers influence the limiting power production (Curzon and Ahlborn 1975; De Vos 1994, Sieniutycz and Kuran 2005, 2006).

Local fluxes of heat and generated power do not change along the process path only when both streams in Fig.1 are infinite. When one, say, upper, stream is finite, its thermal potential decreases along the stream path, which is the consequence of the energy balance. Any finite stream is thus a resource stream. It is the resource property or the finiteness of amount or flow of a valuable substance or energy which leads to changes of the resource stream properties along its Lagrangian path. In the engine mode of the system, one then observes

resource's relaxation to the equilibrium with an infinite lower stream or with an infinite reservoir (usually the environment (Sieniutycz 2009a).

An inverse of the Lagrangian-path-relaxation process (in a consumer-type device) is that in which a resource abandons the equilibrium. That process cannot be spontaneous; it needs a supply of external power. The inverse process may be referred to the thermal or chemical upgrading of the resource, which can be accomplished with a heat pump (Kuran 2006).

From the optimization viewpoint, dynamical is every process with a sequence of states, developing either in the chronological time or in holdup (spatial) time. Studies of resource downgrading or upgrading apply therefore methods of dynamical optimization. A decrease of the resource potentials in an engine or a fuel cell (in chronological or spatial time) with respect to an infinite environment can be studied similarly to an increase of the resource potentials in a heat pump or electrolyser (Berry, Kazakov, Sieniutycz, Szwast and Tsirlin 2000).

2. CONTROLS IN POWER SYSTEMS

Diverse controls can be applied in power systems to represent the propelling fluxes of heat and mass transfer and accomplish the task of a sustainable energy conversion. Here we shall recall and then use definitions of Carnot control variables (Carnot temperature and chemical potential) whose derivations and applications were originated in our previous work (Sieniutycz 2003a,b)

We begin with the simplest case of no mass transfer, i.e. we shall consider a steady, internally reversible ('endoreversible') heat engine with a perfect internal power generator characterized by temperatures of circulating fluid T_1' and T_2' , Fig.1. The stream temperatures, attributed to the bulk of each fluid are T_1 and T_2 . The inequalities $T_1 > T_1' > T_2' > T_2$ are valid for the engine mode of the system.

Evaluating the internal entropy balance of the perfect engine we find

$$\frac{q_2}{T_2'} = \frac{q_1}{T_1'} \quad (1)$$

Continuity of pure heat fluxes through each boundary layer (each conductor) was assumed ($q_1 = q_1'$ and $q_2 = q_2'$), the property which does not hold in the case when heat transfer is coupled with transfer of substances. (As a flux can be normalized by dividing it by a constant resource mass flux we neglect dots over symbols of fluxes).

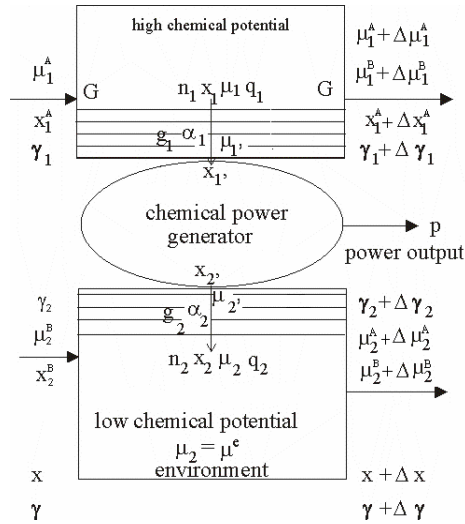


Figure 1. Flow chemical and/or thermal engine

Total entropy balance of the system leads to total entropy source σ_s as the difference between outlet and inlet entropy fluxes

$$\sigma_s = \frac{q_2}{T_2} - \frac{q_1}{T_1} = \frac{q_1}{T_2} \frac{T_2}{T_1} - \frac{q_1}{T_1} = \frac{q_1}{T_2} \left(\frac{T_2}{T_1} - 1 \right). \quad (2)$$

After introducing an effective temperature called Carnot temperature

$$T' \equiv T_2 \frac{T_1}{T_2} \quad (3)$$

entropy production of the endoreversible process, Eq. (2), takes the following simple form

$$\sigma_s = q_1 \left(\frac{1}{T'} - \frac{1}{T_1} \right) \quad (4)$$

This form is identical with the familiar expression obtained for processes of purely dissipative heat exchange between two bodies with temperatures T_1 and T' .

From the entropy and energy balances of an internally reversible ('endoreversible') process the endoreversible thermal efficiency follows in terms of temperatures of the circulating fluid

$$\frac{p}{q_1} = \eta = 1 - \frac{T_2}{T_1} \quad (5)$$

In terms of temperature T' of Eq. (3) the above efficiency assumes the classical Carnot form containing the temperature in the bulk of the second reservoir and the Carnot temperature T'

$$\eta = 1 - \frac{T_2}{T'} \quad (6)$$

This property substantiates the name "Carnot temperature" for control variable T' . When a control action takes place, the superiority of Eq. (6) over Eq. (5) consists in using in (6) single, free control T' , instead of two constrained controls of Eq. (5) (linked by an internal balance of the entropy). Moreover, power

produced in the endoreversible system also takes the classical form

$$p = \eta q_1 = \left(1 - \frac{T_2}{T'} \right) q_1 \quad (7)$$

It is essential that the derivation of Eqs. (1) - (7) does not require any specific assumptions regarding the nature of heat transfer kinetics.

In terms of T' description of thermal endoreversible cycles is broken down to formally "classical" equations which contain T' in place of T_1 . In irreversible situations Carnot temperature T' efficiently represents temperature of the upper reservoir, T_1 . Yet, at the reversible Carnot point, where $T_1' = T_1$ and $T_2' = T_2$, Eq. (3) yields $T' = T_1$, thus returning to the classical reversible theory. These properties of Carnot temperature render descriptions of endoreversible and reversible cycles very similar. They also make the variable T' a suitable control in both static and dynamic situations Sieniutycz (2003a,b).

For the purpose of this paper it is worth knowing that in terms of T' the linear (Newtonian) heat transfer is described by a simple kinetic equation

$$q_1 = g(T_1 - T'), \quad (8)$$

where g is overall heat transfer conductance i.e. the product of a total exchange area and an overall heat transfer coefficient (Berry, Kazakov, Sieniutycz, Szwasz and Tsirlin 2000).

For a linear resource relaxing to the thermodynamic equilibrium along the stationary Lagrangian path (or for an unsteady state relaxation dynamics), the kinetic formula related to Eq. (8) has the linear form

$$\frac{dT_1}{d\tau} = T' - T_1, \quad (9)$$

where the non-dimensional time τ satisfies Eq. (38) below and is related to the overall conductance g of Eq. (8). Subscript 1 will be neglected in equations describing dynamical paths. The resource (or a finite "upper stream") is upgraded whenever Carnot temperature T' is higher than resource's temperature T_1 . Whereas the resource is downgraded (relaxes to the thermodynamic equilibrium with an infinite "lower stream" or the environment of temperature T_2) whenever Carnot temperature T' is lower than resource's temperature T_1 . In linear systems, power-maximizing T' is proportional to the resource's temperature T_1 at each time instant (Sieniutycz 2009a). For the case of two finite streams with constant heat capacities, see Sieniutycz and Jeżowski (2009).

The notion of Carnot temperature can be extended to chemical systems where also the Carnot chemical potential emerges (Sieniutycz 2003b). We shall also make some brief remarks here.

The structure of Eq. (1) also holds to systems with mass transfer provided that instead of pure heat flux q the so called total heat flux (mass transfer involving heat flux) Q is introduced satisfying an equation

$$Q \equiv q + Ts_1 n_1 + \dots + Ts_k n_k + \dots + Ts_m n_m \quad (10)$$

or, since the heat flux equals the difference between total energy flux ε and flux of enthalpies of transferred components, $q = \varepsilon - h$,

$$Q \equiv \varepsilon - \mu_1 n_1 \dots \mu_k n_k \dots - \mu_m n_m \equiv \varepsilon - G \quad (11)$$

where G is the flux of Gibbs thermodynamic function (Gibbs flux). The equality

$$\varepsilon = Q + G \quad (12)$$

is fundamental in the theory of chemical power engines; it indicates that power can be generated by two sorts of propelling fluxes: heat flux Q and Gibbs flux G , each kind of generation having its own efficiency (thermal and chemical efficiencies). The related driving forces are the temperature difference and chemical affinity.

When mass transfer is included the internal entropy balance of the perfect engine has in terms of total heat flux Q the same structure as Eq. (1) in terms of q , i.e.

$$\frac{Q_2}{T_2} = \frac{Q_1}{T_1} \quad (13)$$

The continuity of energy and mass fluxes through the boundary layers leads to 'primed' fluxes in terms of those for the bulk. Assuming a complete conversion we restrict to power yield by a simple reaction $A_1 + A_2 = 0$ (isomerisation or phase change of A_1 into A_2).

The energy balance

$$\varepsilon_1 = \varepsilon_2 + p \quad (14)$$

and the mass balance in terms of conserved fluxes through cross-sections 1' and 1 as well as 2' and 2

$$n_1 = n_2 \quad (15)$$

are combined with Eq. (13) describing the continuity of the entropy flux in the reversible part of the system. This yields

$$\frac{\varepsilon_1 - \mu_1 n_1}{T_1} = \frac{\varepsilon_2 - \mu_2 n_2}{T_2} \quad (16)$$

Eliminating ε_2 and n_2 from these equations yields

$$\frac{\varepsilon_1 - \mu_1 n_1}{T_1} = \frac{\varepsilon_1 - p - \mu_2 n_1}{T_2} \quad (17)$$

whence

$$\frac{p}{T_2} = \frac{\varepsilon_1 - \mu_2 n_1}{T_2} - \frac{\varepsilon_1 - \mu_1 n_1}{T_1} \quad (18)$$

which leads to a power expression

$$p = \varepsilon_1 - \varepsilon_2 = \varepsilon_1 \left(1 - \frac{T_2}{T_1}\right) + T_2 \left(\frac{\mu_1}{T_1} - \frac{\mu_2}{T_2}\right) n_1 \quad (19)$$

In Eq. (19) power p is expressed in terms of the fluxes continuous through the conductors.

Entropy production in the system follows from the balance of fluxes in the bulks of the streams

$$\sigma_s = \frac{q_2}{T_2} - \frac{q_1}{T_1} + (s_2 - s_1) n_1 \quad (20)$$

Eliminating q_2 from this result with the help of the energy balance (14) we obtain

$$\sigma_s = (q_1 + h_1 n_1) \left(\frac{1}{T_2} - \frac{1}{T_1}\right) + \left(\frac{\mu_1}{T_1} - \frac{\mu_2}{T_2}\right) n_1 - \frac{p}{T_2} \quad (21)$$

An equivalent form of this equation is the formula

$$p = \varepsilon_1 \left(1 - \frac{T_2}{T_1}\right) + T_2 \left(\frac{\mu_1}{T_1} - \frac{\mu_2}{T_2}\right) n_1 - T_2 \sigma_s \quad (22)$$

which may be compared with the same power evaluated for the endoreversible part of the system

$$p = \varepsilon_1 \left(1 - \frac{T_2'}{T_1'}\right) + T_2' \left(\frac{\mu_1'}{T_1'} - \frac{\mu_2'}{T_2'}\right) n_1 \quad (23)$$

The comparison of Eqs (22) and (23) yields an equality

$$\begin{aligned} & \varepsilon_1 \left(1 - \frac{T_2}{T_1}\right) + T_2 \left(\frac{\mu_1}{T_1} - \frac{\mu_2}{T_2}\right) n_1 - T_2 \sigma_s \\ &= \varepsilon_1 \left(1 - \frac{T_2'}{T_1'}\right) + T_2' \left(\frac{\mu_1'}{T_1'} - \frac{\mu_2'}{T_2'}\right) n_1 \end{aligned} \quad (24)$$

from which the entropy production can be expressed in terms of bulk driving forces and active driving forces (measures of process efficiencies).

We finally obtain

$$\begin{aligned} \sigma_s &= \frac{\varepsilon_1}{T_2} \left(\frac{T_2'}{T_1'} - \frac{T_2}{T_1}\right) \\ &+ n_1 \left(\frac{\mu_1}{T_1} - \frac{T_2'}{T_2} \left(\frac{\mu_1'}{T_1'} - \frac{\mu_2'}{T_2'}\right) - \frac{\mu_2}{T_2}\right) \end{aligned} \quad (25)$$

This expression generalizes Eq. (3) for the case when a single reaction $A_1 + A_2 = 0$ undergoes in the system. Equation (25) leads again to the definition of Carnot temperature in agreement with Eq. (3) and to Carnot chemical potential of the first component

$$\frac{\mu'}{T'} = \frac{\mu_2}{T_2} + \frac{T_2'}{T_2} \left(\frac{\mu_1'}{T_1'} - \frac{\mu_2'}{T_2'}\right) \quad (26)$$

In a special case of an isothermal process the above formula yields a chemical control variable

$$\mu' = \mu_2 + \mu_1 - \mu_2 \quad (27)$$

which has been used earlier to study an isothermal engine (Sieniutycz 2008). After introducing the Carnot temperature in accordance with Eq. (3), total entropy production of the endoreversible power generation by the simple reaction $A_1 + A_2 = 0$ (isomerisation or phase change of A_1 into A_2), takes the following simple form

$$\sigma_s = \varepsilon_1 \left(\frac{1}{T'} - \frac{1}{T_1}\right) + \left(\frac{\mu_1}{T_1} - \frac{\mu'}{T'}\right) n_1 \quad (28)$$

Introducing the above formula total heat Q_1 satisfying $Q_1 \equiv \varepsilon_1 - \mu_1 n_1$ we finally obtain

$$\sigma_s = Q_1 \left(\frac{1}{T'} - \frac{1}{T_1}\right) + n_1 \frac{\mu_1 - \mu'}{T'} \quad (29)$$

where $Q_1 = q_1 + T_1 s_1 n_1$ is the total heat flux propelling the power generation in the system. The resulting equation is formally equivalent with an expression obtained for a process of purely dissipative exchange of energy and matter between two bodies with temperatures T_1 and T' and chemical potentials μ_1 and μ' .

Carnot variables T' and μ' are two free, independent control variables applied in power maximization of steady and dynamical generators.

3. STEADY POWER SYSTEMS

The majority of research papers on power limits published to date deals with systems in which there are

two infinite reservoirs. To this case refer the steady-state analyses of the Chambadal-Novikov-Curzon-Ahlborn engine (CNCA engine) in which energy exchange is described by Newtonian law of cooling, Curzon and Ahlborn 1975, or of the Stefan-Boltzmann engine, a system with the radiation fluids and the energy exchange governed by the Stefan-Boltzmann law (De Vos 1994). The entropy production and power yield characteristics for these systems are shown in Fig. 2.

In a CNCA engine the maximum power point may be related to the optimum value of a free (unconstrained) control variable which can be efficiency η , heat flux q_1 , or Carnot temperature T' . When internal irreversibilities within the power generator play a role, a pseudo-Carnot formula $\eta = 1 - \Phi T_2/T'$, applies in place of Eq. (6), where Φ is the internal irreversibility factor (Sieniutycz and Kuran 2006). In terms of bulk temperatures T_1, T_2 and Φ one finds $T'_{opt} = (T_1 \Phi T_2)^{1/2}$.

For the Stefan-Boltzmann engine exact expression for the optimal point cannot be determined analytically, yet, this temperature can be found graphically from the chart $p=f(T')$. Yet, the so-called pseudo-Newtonian model, Sieniutycz and Kuran 2006, Kuran 2006, which uses state dependent heat exchange coefficient, $\alpha(T^3)$, omits, to a considerable extent, analytical difficulties associated with the Stefan-Boltzmann equation.

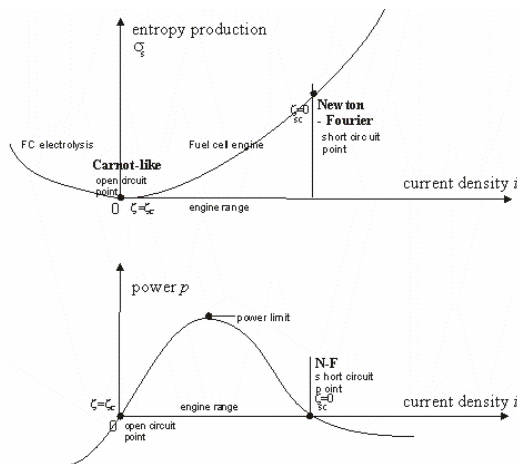


Figure 2. Qualitative picture illustrating entropy production and power yield in fuel cells in terms of the density of electric current. For thermal engines the picture is qualitatively similar provided that the electric current is replaced by the entropy flux

4. DYNAMICAL POWER SYSTEMS

Dynamical energy yield requires the knowledge of an extremal curve rather than an extremum point. This refers to variational methods (to handle functional extrema) in place of static optimization methods (to handle extrema of functions). For example, the use of a pseudo-Newtonian model to quantify the power yield from radiation, gives rise to a non-exponential optimal curve describing the Lagrangian path of the radiation relaxation to the equilibrium. The shape of the curve is the consequence of

nonlinear properties of the relaxation dynamics. Non-exponential are also other curves describing the radiation relaxation, e.g. those following from exact models using the Stefan-Boltzmann equation (Kuran 2006, Sieniutycz and Kuran 2005, 2006). Optimal (e.g. power-maximizing) relaxation state $T(t)$ is associated with the optimal control $T'(t)$; they both are components of the dynamic optimization solution.

Energy limits of dynamical processes are inherently connected with the exergies, the classical exergy and its rate-dependent extensions. To obtain the classical exergy from power functions it suffices to assume that the thermal efficiency of the system is identical with the Carnot efficiency. On the other hand, non-Carnot efficiencies lead to 'generalized exergies'. The benefit from generalized exergies is that they define stronger energy limits than those predicted by classical exergies.

5. FINITE RATE EXERGIES

Two different kinds of work, first associated with the resource downgrading during its relaxation to the equilibrium and the second – with the reverse process of resource upgrading, are essential. During the approach to the equilibrium engine mode takes place in which work is released, during the departure- heat-pump mode occurs in which work is supplied. Work W delivered in the engine mode is positive by assumption ("engine convention").

The optimal work is sought in the form of a generalized potential that depends on the end states and duration. For appropriate boundary conditions the principal function of the variational problem of extremum work at flow coincides with the exergy as the function that characterizes quality of resources.

Total power obtained from an infinite number of infinitesimal stages representing the relaxation process is determined as the Lagrange functional of the following structure

$$\dot{W}[\mathbf{T}^i, \mathbf{T}^f] = \int_{t^i}^{t^f} f_0(T, T') dt = - \int_{t^i}^{t^f} \dot{G}c(T)\eta(T, T') \dot{T} dt \quad (30)$$

where f_0 is power generation intensity, \dot{G} - resource flux, $c(T)$ -specific heat, $\eta(T, T')$ - efficiency in terms of state T and control T' , further \mathbf{T} - enlarged state vector comprising state and time, t - time variable (residence time or holdup time) for a resource contacting with heat transfer surface. A non-dimensional time τ is used in the relaxation description

$$\tau \equiv \frac{x}{H_{TU}} = \frac{\alpha' a_v F}{\dot{G}c} x = \frac{\alpha' a_v F v}{\dot{G}c} t = \frac{t}{\chi} \quad (31)$$

The above definition assures that τ is identical with the number of the energy transfer units, and related to the system's time constants, χ and H_{TU} (relaxation constant and height of the transfer unit). Equation (31) refers to resource's flow \dot{G} , stream velocity v through cross-section A^\perp , and the heat transfer exchange surface per unit volume a_v (Sieniutycz and Kuran 2006).

For a constant mass flux of a resource stream, one can extremize power per unit mass flux, i.e. the quantity

of specific work dimension called ‘work at flow’. The function f_0 in Eq. (30) contains thermal efficiency, η , described by a practical counterpart of the Carnot formula. When $T > T^e$, efficiency η decreases in the engine mode above η_C and increases in the heat-pump mode below η_C . At the limit of vanishing rates $dT/dt = 0$ and $T' \rightarrow T$. Work of each mode simplifies then to the classical exergy.

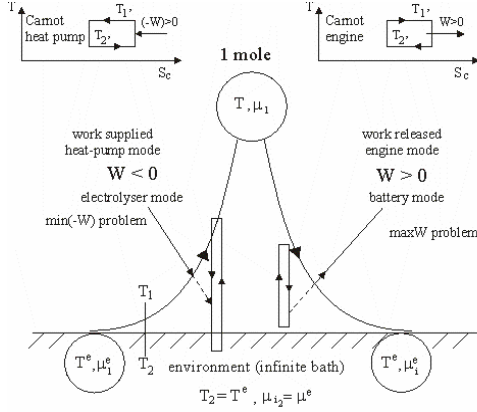


Figure 3. In finite-rate processes limiting work produced and consumed differ in both process modes.

Solutions to work extremum problems can be obtained by variational methods, i.e. via Euler-Lagrange equation of variational calculus. However, the solution of Euler-Lagrange equation does not contain direct information about the optimal work function $V = \max(\dot{W} / \dot{G})$. Yet, this function can be obtained by solving the related Hamilton-Jacobi-Bellman equation (HJB equation: Bellman 1961; Berry, Kazakov, Sieniutycz, Szwasz and Tsirlin 2000). For the linear kinetics

$$\frac{\partial V}{\partial \tau} - \max_{T'} \left\{ -\frac{\partial V}{\partial T} - c \left(1 - \frac{T^e}{T'} \right) (T' - T) \right\} = 0 \quad (32)$$

The extremal work function $V = \max(\dot{W} / \dot{G})$ is a function of the final state and total duration. After evaluation of optimal control and its substitution to Eq. (32) one obtains a nonlinear equation

$$\frac{\partial V}{\partial \tau} - c \left\{ \sqrt{T^e} - \sqrt{T(1 + c^{-1} \partial V / \partial T)} \right\}^2 = 0 \quad (33)$$

which is the Hamilton-Jacobi equation of the problem. Its solution can be found by the integration of work intensity along an optimal path, between limits T^i and T^f . A reversible (path independent) part of V is the classical exergy $A(T, T^e, 0)$. In the case of analytical difficulties the method of dynamic programming is applied to solve a discrete HJB equation which is in, fact, Belman's equation of dynamic programming describing a multistage cascade process. Details of modeling of multistage power production in sequences of engines are discussed in the previous publications (Sieniutycz and Kuran 2006, Sieniutycz and Jeżowski 2009).

6. EXAMPLES OF HJB EQUATIONS FOR NONLINEAR THERMAL SYSTEMS

We shall display here some Hamilton-Jacobi-Bellman equations for radiation power systems. A suitable example is a radiation engine whose power integral is approximated by a pseudo-Newtonian model of radiative energy exchange. The model is associated with an optimal function

$$V(T^i, t^i, T^f, t^f) \equiv \max_{T'(t)} \left(- \int_{t^i}^{t^f} \dot{G}_m c_m \left(1 - \Phi' \frac{T^e}{T'} \right) v(T', T) dt \right) \quad (34)$$

where $v = \alpha(T^3)(T' - T)$. Alternative forms use expressions of Carnot temperature T' in terms of other control variables (Sieniutycz and Kuran 2006). Optimal power (34) can be referred to a pseudolinear kinetics $dT/dt = f(T, T')$ consistent with rate $v = \alpha(T^3)(T' - T)$. A general form of HJB equation for work function V is

$$-\frac{\partial V}{\partial t} + \max_{T'(t)} \left(f_0(T, T') - \frac{\partial V}{\partial T} f(T, T') \right) = 0, \quad (35)$$

where f_0 is defined as the integrand in Eq. (34).

A more exact model of radiation conversion relaxes the assumption of the pseudo-Newtonian transfer and applies the Stefan-Boltzmann law. For a *symmetric* model of radiation conversion (both reservoirs composed of radiation

$$\dot{W} = \int_{t^i}^{t^f} \dot{G}_c(T) \left(1 - \frac{\Phi T^e}{T'} \right) \beta \frac{T^a - T'^a}{(\Phi'(T'/T^e)^{a-1} + 1) T^{a-1}} dt. \quad (36)$$

The coefficient $\beta = \sigma a_v c_h^{-1} (p_m^0)^{-1}$ is related to molar constant of photons density p_m^0 and Stefan-Boltzmann constant σ . In the physical space, power exponent $a=4$ for radiation and $a=1$ for a linear resource. With the dynamical state equation

$$\frac{dT}{dt} = -\beta \frac{T^a - T'^a}{(\Phi'(T'/T^e)^{a-1} + 1) T^{a-1}} \quad (37)$$

applied in general Eq. (35) we obtain a HJB equation

$$\frac{\partial V}{\partial t} = \max_{T'(t)} \left\{ \dot{G}_c \left(1 - \Phi \frac{T^e}{T'} \right) + \frac{\partial V}{\partial T} \beta \frac{T^a - T'^a}{(\Phi'(T'/T^e)^{a-1} + 1) T^{a-1}} \right\} \quad (38)$$

(Sieniutycz and Kuran 2006). Dynamics (37) is the characteristic equation to Eq. (38).

For a *hybrid model* of radiation conversion (upper reservoir composed of the radiation and lower reservoir of a Newtonian fluid, power is

$$\dot{W} = - \int_{\tau^i}^{\tau^f} G_c(T) \left(1 - \frac{\Phi T^e}{T'} \right) u dt \quad (39)$$

whereas the corresponding Hamilton-Jacobi-Bellman equation has the form

$$-\frac{\partial V}{\partial t^f} + \max_{T'(t)} \left\{ - \left(\dot{G}_c(T) \left(1 - \frac{\Phi T^e}{T'} \right) + \frac{\partial V}{\partial T^f} \right) u \right\} = 0 \quad (40)$$

where by definition:

$$T' \equiv (T^a + \beta^{-1} T^{a-1} u)^{1/a} + \Phi \beta^{-1} T^{a-1} u g_1 / g_2$$

is Carnot temperature of this particular problem (Sieniutycz and Kuran 2006).

7. SOLUTIONS OF HJB EQUATIONS FOR THERMAL POWER SYSTEMS

In all HJB equations extremized expressions are some hamiltonians. By feedback control optimal driving temperature T' or some other control is implemented as the quantity maximizing the hamiltonian with respect to T' at each point of the path. The maximization of H leads to two equations. The first expresses optimal control T' in terms of T and $z = -\partial V/\partial T$. For the linear kinetics of Eq. (32) we obtain

$$\frac{\partial V}{\partial T} - \frac{\partial f_0(T, T')}{\partial T'} = \frac{\partial V}{\partial T} + c(1 - \frac{T^e T}{T'^2}) = 0 \quad (41)$$

whereas the second is the original equation (32) without maximizing operation

$$\frac{\partial V}{\partial \tau} + \frac{\partial V}{\partial T}(T' - T) + c(1 - \frac{T^e}{T'})(T' - T) = 0 \quad (42)$$

The Pontryagin's variable for this problem is $z = -\partial V/\partial T$. To obtain optimal control function $T'(z, T)$ one should solve the second equality in equation (41) in terms of T' . The result is optimal Carnot control T' in terms of T and $z = -\partial V/\partial T$,

$$T' = \left(\frac{T^e T}{1 + c^{-1} \partial V / \partial T} \right)^{1/2}. \quad (43)$$

This expression is next substituted into (42); the result is the nonlinear Hamilton-Jacobi equation

$$-\frac{\partial V}{\partial \tau} + cT \left(\sqrt{1 + c^{-1} \partial V / \partial T} - \sqrt{T^e / T} \right)^2 = 0 \quad (44)$$

which contains the energylike (extremum) Hamiltonian of the extremum process

$$H(T, \frac{\partial V}{\partial T}) = cT \left(\sqrt{1 + c^{-1} \partial V / \partial T} - \sqrt{T^e / T} \right)^2. \quad (45)$$

For a positively-defined H , each Hamilton-Jacobi equation for optimal work preserves the general form of autonomous equations known from analytical mechanics and theory of optimal control.

Expressing extremum Hamiltonian (45) in terms of state variable T and Carnot control T' yields an energylike function satisfying the following relation

$$E(T, u) = f_0 - u \frac{\partial f_0}{\partial u} = cT^e \frac{(T' - T)^2}{T'^2} \quad (46)$$

E is the Legendre transform of the work lagrangian $l_0 = -f_0$ with respect to the rate $u = dT/d\tau$.

Assuming a value of the Hamiltonian, say h , one can exploit the constancy of H to eliminate $\partial V/\partial T$. Next combining equation $H=h$ with optimal control (52), or with an equivalent result for heat flow control $u=T'-T$

$$u = \left(\frac{T^e T}{1 + c^{-1} \partial V / \partial T} \right)^{1/2} - T \quad (47)$$

yields optimal rate $u = \dot{T}$ in terms of temperature T and the Hamiltonian constant h

$$\dot{T} = \{ \pm \sqrt{h / cT^e} (1 \pm \sqrt{h / cT^e})^{-1} \} T. \quad (48)$$

A more general form of this result which applies to systems with internal dissipation (factor Φ) and applies to the pseudo-Newtonian model of radiation is

$$\dot{T} = \left(\pm \sqrt{\frac{h_\sigma}{\Phi c_v(T)}} \left(1 \pm \sqrt{\frac{h_\sigma}{\Phi c_v(T)}} \right)^{-1} \right) T \equiv \xi(h_\sigma, \Phi, T) T \quad (49)$$

where ξ , defined in the above equation, is an intensity index and $h_\sigma = h/T$. This result is valid the temperature dependent heat capacity $c_v(T) = 4a_0 T^3$. Positive ξ refer to heating of the resource fluid in the heat-pump mode, and the negative - to cooling of this fluid in the engine mode. Thus pseudo-Newtonian systems produce power relaxing with the optimal rate

$$\dot{T} = \xi(h_\sigma, T, \Phi) T. \quad (50)$$

Equations (49) and (50) describe the optimal trajectory in terms of state variable T and constant h_σ . The corresponding optimal control (Carnot control) is

$$T' = (1 + \xi(h_\sigma, \Phi, T)) T. \quad (51)$$

In comparison with the linear systems, the pseudo-Newtonian relaxation curve is not exponential. Kuran (2006) has illustrated the optimal temperature of radiation downgraded in engine mode or upgraded in the heat-pump mode (Kuran 2006; Sieniutycz and Kuran 2005, 2006).

8. EXAMPLES OF OPTIMAL WORK FUNCTIONS

Let us begin with linear systems. Substituting temperature control (51) with a constant ξ into work functional (31) and integrating along an optimal path yields an extremal work function

$$V(T^i, T^f, h) = c(T^i - T^f) - cT^e \ln \frac{T^i}{T^f} \quad (52)$$

$$- cT^e \sqrt{\frac{h}{cT^e}} \ln \frac{T^i}{T^f}$$

This expression is valid for every process mode. Integration of Eq (49) subject to end conditions $T(\tau^i) = T^i$ and $T(\tau^f) = T^f$ leads to V in terms of the process duration.

For radiation $c_v(T) = 4a_0 T^3$, where a_0 is the radiation constant. The optimal path solving Eqs. (57) and (59) satisfies an equation

$$\pm (4/3) a_0^{1/2} \Phi^{1/2} h_\sigma^{-1/2} \left(T^{3/2} - T^{i3/2} \right). \quad (53)$$

$$-\ln(T/T^i) = \tau - \tau^i$$

The integration limits refer to the initial state (i) and a current state of the radiation fluid, i.e. temperatures T^i and T corresponding with τ^i and τ . Optimal curve (53) refers to the case when the radiation relaxation is subject to a constraint resulting from Eq. (50).

Equation (53) is associated with an entropy production term for Eq. (34). The corresponding extremal work function per unit volume of flowing radiation is

$$\begin{aligned}
V &\equiv h_v^i - h_v^f - T^e (s_v^i - s_v^f) \\
&- (4/3)a_0^{1/2} h_\sigma^{1/2} \Phi^{1/2} T^e (T^{i3/2} - T^{f3/2}) \\
&+ (4/3)a_0 T^e (1 - \Phi)(T^{i3} - T^{f3})
\end{aligned} \quad (54)$$

The generalized exergy of radiation at flow (Petela 1964) follows in analytical form from Eq. (54) after applying exergy boundary conditions. Yet the classical exergy of radiation at flow resides in the resulting exergy equation in Jeter's (1981) rather than in Petela's (1964) form.

9. CHEMICAL POWER SYSTEMS

The developed approach can be extended to chemical and electrochemical engines. Here we shall make only a few basic remarks. In chemical engines mass transports participate in transformation of chemical affinities into mechanical power (Tsirlin, Kazakov, Mironova and Amelkin 1998; Sieniutycz 2008). Yet, as opposed to thermal machines, in chemical ones generalized streams or reservoirs are present, capable of providing both heat and substance. Large streams or infinite reservoirs assure constancy of chemical potentials. Problems of extremum power (maximum of power produced and minimum of power consumed) are static optimization problems. For a finite "upper stream", however, amount and chemical potential of an active reactant decrease in time, and considered problems are those of dynamic optimization and variational calculus. Because of the diversity and complexity of chemical systems the area of power producing chemistries is extremely broad.

The simplest model of power producing chemical engine is that with an isothermal isomerization reaction, $A_1 + A_2 = 0$ (De Vos 1994; Sieniutycz 2008). Power expression and efficiency formula for the chemical system follow from the entropy conservation and energy balance in the power-producing zone of the system ('active part'). In an 'endoreversible chemical engine' total entropy flux is continuous through the active zone. When a formula describing this continuity is combined with energy balance we find in an isothermal case

$$p = (\mu_1 - \mu_2)n_1, \quad (55)$$

where the feed flux n_1 equals to n , an invariant molar flux of reagents. Process efficiency ζ is defined as power yield per molar flux, n . This efficiency is identical with the chemical affinity of our reaction in the chemically active part of the system. While ζ is not dimensionless, it describes correctly the system. In terms of Carnot variable, μ' , which satisfies Eq. (31),

$$\zeta = \mu' - \mu_2. \quad (56)$$

For a steady engine the following function describes chemical Carnot control μ' in terms of fuel flux n_1 and its mole fraction x

$$\mu' = \mu_2 + \zeta_0 + RT \ln \left(\frac{x_1 - n_1 g_1^{-1}}{n_1 g_2^{-1} + x_2} \right) \quad (57)$$

As Eq. (56) is valid, Eq. (57) also characterizes the efficiency control in terms of n and fuel fraction x .

Equation (57) shows that an effective concentration of the reactant in upper reservoir $x_{1\text{eff}} = x_1 - g_1^{-1}n$ is decreased, whereas an effective concentration of the product in lower reservoir $x_{2\text{eff}} = x_2 + g_2^{-1}n$ is increased due to the finite mass flux. Therefore efficiency ζ decreases nonlinearly with n . When effect of resistances g_k^{-1} is ignorable or flux n is very small, reversible Carnot-like chemical efficiency, ζ_c , is attained. The power function, described by the product $\zeta(n)n$, exhibits a maximum for a finite value of the fuel flux, n .

Application of Eq. (57) to the Lagrangian relaxation path leads to a work functional

$$W = - \int_{\tau_1^i}^{\tau_1^f} \left\{ \zeta_0 + RT \ln \left(\frac{X/(1+X) + dX/d\tau_1}{x_2 - j dX/d\tau_1} \right) \right\} \frac{dX}{d\tau_1} d\tau_1 \quad (58)$$

whose maximum describes the dynamical limit of the system. Here $X = x/(1-x)$ and j equals the ratio of upper to lower mass conductance, g_1/g_2 . The path optimality condition may be expressed in terms of the constancy of the following Hamiltonian

$$H(X, \dot{X}) = RT \dot{X}^2 \left(\frac{1+X}{X} + \frac{j}{x_2} \right). \quad (59)$$

For low rates and large concentrations X (mole fractions x_1 close to the unity) optimal relaxation rate of the fuel resource is approximately constant. Yet, in an arbitrary situation optimal rates are state dependent so as to preserve the constancy of H in Eq. (59). Extensions of Eq. (57) are available for multicomponent and multireaction systems (Sieniutycz 2009b).

10. STEADY FUEL CELLS

To understand the role of chemical reactions in the power yield we consider performance bounds of fuel cells. These systems are electrochemical flow engines propelled by chemical reactions, which satisfy requirements imposed by chemical stoichiometry. The performance of fuel cells is determined by magnitudes and directions of all streams and by mechanism of electric current generation. The mode distinction for the work production and consumption units applies here as well. Units which produce power are electrochemical engines whereas those which consume power are electrolyzers. Figure 4 below illustrates a solid oxide fuel cell engine (SOFC) and refers to the power yield mode.

A fuel cell is an electrochemical energy converter which directly and continuously transforms a part of chemical energy into electrical energy by consuming fuel and oxidant. Fuel cells have recently attracted great attention by virtue of their inherently clean, efficient, and reliable performance. Their main advantage as compared to heat engines is that their efficiency is not a major function of device size.

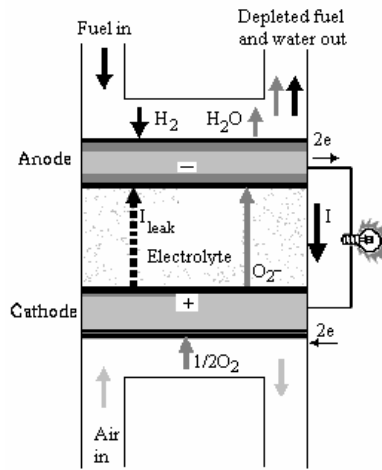


Figure 4: Principle of a solid oxide fuel cell

While both electronic and ionic transfers are necessary to sustain the power generation, it is the overall chemical reaction which is the source of power, and it is the chemical unit property which constitutes the first major component of the theory of power generation in fuel cell engines. The second major component involves the kinetics of electronic, ionic and thermal transfer phenomena.

The basic structure of fuel cells includes electrolyte layer in contact with a porous anode and cathode on either side. Gaseous fuels are fed continuously to the anode (negative electrode) compartment and an oxidant (i.e., oxygen from air) is fed continuously to the cathode (positive electrode) compartment. The electrochemical reactions take place at the electrodes to produce an electric current. The reaction is the electrochemical oxidation of fuel, usually hydrogen, and the reduction of the oxidant, usually oxygen. This principle makes fuel cells similar to a chemical engine of Fig. 1.

Voltage lowering in fuel cells below the reversible value is a good measure of their imperfection, Fig.5. With the concept of effective nonlinear resistances operating voltage of a fuel cell can be represented as the departure from the ideal voltage E

$$V = E - V_{\text{int}} = E - V_{\text{act}} - V_{\text{conc}} - V_{\text{ohm}} = E - I(R_{\text{act}} + R_{\text{conc}} + R_{\text{ohm}}) \quad (60)$$

The losses, which are called polarization, include three main sources: activation polarization (V_{act}), ohmic polarization (V_{ohm}), and concentration polarization (V_{conc}). They refer to the equivalent activation resistance (R_{act}), equivalent ohmic resistance (R_{ohm}), and equivalent concentration resistance (R_{conc}). Large number of approaches for calculating these polarization losses has been presented in literature (Zhao, Ou and Chen 2008). Activation and concentration polarization occurs at both anode and cathode locations, while the resistive polarization represents ohmic losses throughout the fuel cell.

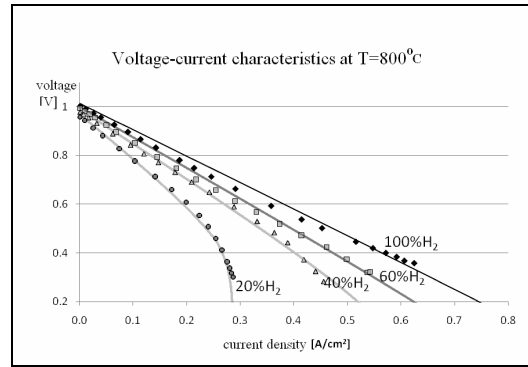


Figure 5. Voltage vs current characteristics of the SOFC in temperature 800°C. Continuous lines represent the Aspen Plus™ calculations testing the model consistency with the experiments. These lines were obtained in Wierzbicki's MsD thesis supervised by the present author and J. Jewulski (Wierzbicki 2009). Points refer to experiments of Wierzbicki and Jewulski in Warsaw Institute of Energetics (Wierzbicki 2009 and his ref 18).

In the literature there are many experimental and theoretical examples showing power maxima in fuel cells and proving the suitability of the thermal machine theory to chemical and electrochemical systems.

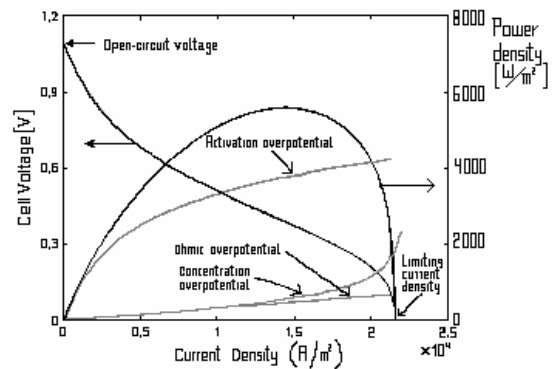


Figure 6. Data of the cell voltage, polarizations, and power density in terms of current density for a fuel cell using hydrogen (97% H_2 + 3% H_2O) as fuel and air (21% O_2 + 79% N_2) as oxidant (Zhao, Ou and Chen 2008), consistent with the data of Wierzbicki (2009).

11. CONCLUDING REMARKS

Our research provides data for power production bounds (limits) which are enhanced in comparison with bounds predicted by the classical thermodynamics. As opposed to the classical thermodynamics, in general, these bounds depend not only on changes of the thermodynamic state of participating resources but also on process irreversibilities, ratios of stream flows, stream directions, and mechanism of heat and mass transfer. The methodology known for thermal machines has been extended to chemical and electrochemical engines. Extensions are also available for multicomponent, multireaction units (Sieniutycz 2009b)

To understand the problem of bounds and their distinction for the work production and consumption,

recall that the work-producing process is the inverse of the work-consuming process (the final state of the second process is the initial state of the first, and conversely), when the durations of the two processes and their end states are fixed to be the same. In thermostatics the two bounds on the work, the bound on the work produced and that on the work consumed, coincide. However the thermostatic bounds are often too far from reality to be really useful. The generalized bounds, obtained here by solving HJB equations for optimal Lagrangian paths, are stronger than those predicted by thermostatics. They do not coincide for processes of work production and work consumption; they are 'thermokinetic' rather than 'thermostatic' bounds. Only for infinitely long durations or for processes with excellent transfer (an infinite number of transfer units) the thermokinetic bounds reduce to the classical thermostatic bounds.

A real process which does not apply the optimal protocol but has the same boundary states and duration as the optimal Lagrangian path, requires a real work supply that can only be larger than the finite-rate limit obtained by the optimization. Similarly, the real work delivered from a nonequilibrium work-producing system (with the same boundary states and duration but with a suboptimal control) can only be lower than the corresponding finite-rate limit. Indeed, the two bounds, for a process and its inverse, which coincide in thermostatics, diverge in thermodynamics, at a rate that grows with any index quantifying the process deviation from the static behavior, e.g. Hamiltonian H . For sufficiently high values of rate indices (large H), work consumed may far exceed the classical work; work produced can be much lower than classical or even vanish. Thus, with thermokinetic models, we can confront and surmount the limitations of applying classical thermodynamic bounds to real processes. This is a direction with many open opportunities, especially for separation and chemical systems. Electrochemical systems and particularly fuel cells are especially important in this context by virtue of their inherently clean, efficient, and reliable performance.

ACKNOWLEDGMENTS

This research was supported in part by Polish Ministry of Science, grant NN208 019434: Thermodynamics and Optimization of Chemical and Electrochemical Energy Generators with Applications to Fuel Cells. The results in Fig. 5 were obtained with M. Wierzbicki during the present author's supervising of his MsD thesis. The thesis co-supervising by dr J. Jewulski of the Warsaw Institute of Energetics is also appreciated. Fig. 6 shows the results of Lingen Chen's research group, consistent with those of M. Wierzbicki (Zhao, Ou and Chen 2008).

APPENDIX A: NOMENCLATURE

A_v generalized exergy per unit volume [Jm^{-3}]
 A^\perp surface area perpendicular to flow [m^2]
 a temperature power exponent in kinetic equation [-]
 $a_0=4\sigma/c$ constant related to the Stefan-Boltzmann constant [$\text{Jm}^{-3}\text{K}^{-4}$]
 a_v total area of energy exchange per unit volume [m^{-1}]

\dot{G} resource flux [gs^{-1} , mols^{-1}]
 g_1, g partial and overall conductance [$\text{Js}^{-1}\text{K}^{-a}$]
 f_0, f_i profit rate and process rates
 H Hamiltonian function
 H_{TU} height of transfer unit [m]
 h numerical value of Hamiltonian [$\text{Jm}^{-3}\text{K}^{-1}$]
 h, h_v specific and volumetric enthalpies [Jg^{-1} , Jm^{-3}]
 i -electric current density [Am^{-2}]
 n flux of fuel reagents [gs^{-1} , mols^{-1}]
 $p = \dot{W}$ power output [Js^{-1}]
 p_m^0 molar constant of photons density [$\text{molm}^{-2}\text{K}^{-3}\text{s}^{-1}$]
 q heat flux between a stream and power generator [Js^{-1}]
 Q total heat flux involving transferred entropies [Js^{-1}]
 S, S_σ entropy and entropy produced [JK^{-1}]
 s, s_v specific and volumetric entropy [$\text{JK}^{-1}\text{g}^{-1}$, $\text{JK}^{-1}\text{m}^{-3}$]
 T variable temperature of resource [K]
 T_1, T_2 bulk temperatures of reservoirs 1 and 2 [K]
 T_1, T_2 temperatures of circulating fluid (Fig.1) [K]
 T^e constant temperature of environment [K]
 T' Carnot temperature control [K],
 $\dot{T} = u$ rate of control of T in non-dimensional time [K]
 t physical time [s]
 u and v rate controls, $dT/d\tau$ and dT/dt , [K , Ks^{-1}]
 V voltage, maximum work function, resp. [V , Jmol^{-1}]
 v velocity of resource stream [ms^{-1}]
 W work produced, positive in engine mode [J]
 w specific work at flow or power per unit flux of a resource [J/mol]
 x mass fraction [-], length coordinate [m]
 z adjoint variable

Greek symbols

α_1, α' partial and overall heat coefficients referred to respective cross-sections [$\text{Jm}^{-2}\text{s}^{-1}\text{K}^{-1}$]
 β effective coefficient of radiation transfer related to molar constant of photons density p_m^0 and Stefan-Boltzmann constant of radiation; $\beta = \alpha_{v,c} h^{-1} (p_m^0)^{-1}$ [s^{-1}]
 ε total energy flux, conservative along a conductor [Js^{-1}]
 $\eta = p/q_1$ first-law thermal efficiency [-]
 $\chi = \rho c_v (\alpha' a_v)^{-1}$ time constant assuring the identity of ratio t/χ with number of transfer units [s]
 μ chemical potential [Jmol^{-1}]
 μ' Carnot chemical potential [Jmol^{-1}]
 Φ factor of internal irreversibility [-]
 σ Stefan-Boltzmann constant for radiation [$\text{Jm}^{-2}\text{s}^{-1}\text{K}^{-4}$]
 σ_s entropy production of the system [$\text{JK}^{-1}\text{s}^{-1}$]
 ξ intensity index [-]
 ζ chemical efficiency [-]
 τ dimensionless time or number of transfer units [-]

Subscripts

C Carnot point
 m molar flow
 v per unit volume
 $1, 2$ first and second fluid

Superscripts

e environment

i initial state

f initial state

Abbreviations

CNCA Chambadal-Novikov-Curzon-Ahlborn engine

HJB Hamilton-Jacobi-Bellman

HJ Hamilton Jacobi equation.

REFERENCES

- Bellman, R., 1961. *Adaptive Control Processes: a Guided Tour*. Princeton University Press.
- Berry, R. S., Kazakov, V. A., Sieniutycz, S., Szwasz, Z., Tsirlin, A. M., 2000. *Thermodynamic Optimization of Finite Time Processes*, Chichester, Wiley, p.197.
- Curzon, F.L., Ahlborn, B., 1975. Efficiency of Carnot engine at maximum power output. *American J. Phys.*, 43(1), 22-24.
- De Vos, A., 1994. *Endoreversible Thermodynamics of Solar Energy Conversion*, Oxford University Press, pp. 30-41.
- Jeter, J., 1981. Maximum conversion efficiency for the utilization of direct solar radiation. *Solar Energy*, 26(3), 231-236.
- Kuran, P., 2006. *Nonlinear Models of Production of Mechanical Energy in Non-Ideal Generators Driven by Thermal or Solar Energy*, Thesis (PhD). Warsaw University of Technology.
- Petela, R., 1964. Exergy of heat radiation. *J. Heat Transfer*, 86(2), 187-192.
- Sieniutycz, S., 2003a. A synthesis of thermodynamic models unifying traditional and work-driven operations with heat and mass exchange. *Open Sys. & Information Dynamics*, 10(1), 31-49.
- Sieniutycz, S., 2003b. Carnot controls to unify traditional and work-assisted operations with heat & mass transfer. *International Journal of Applied Thermodynamics*, 6(2), 59-67.
- Sieniutycz, S., 2008. An analysis of power and entropy generation in a chemical engine. *Intern. J. of Heat and Mass Transfer*, 51(25-26), 5859-5871.
- Sieniutycz, S., 2009a, Dynamic programming and Lagrange multipliers for active relaxation of resources in non-equilibrium systems. *Applied Mathematical Modeling*, 33(3), 1457-1478.
- Sieniutycz, S., 2009b. Complex chemical systems with power production driven by mass transfer. *Intern. J. of Heat and Mass Transfer*, 52(10), 2453-2465.
- Sieniutycz, S., Jeżowski, J., 2009. *Energy Optimization in Process Systems*. Oxford: Elsevier.
- Sieniutycz, S., Kuran, P., 2005. Nonlinear models for mechanical energy production in imperfect generators driven by thermal or solar energy, *Intern. J. Heat Mass Transfer*, 48(3-4), 719-730.
- Sieniutycz, S., Kuran, P., 2006. Modeling thermal behavior and work flux in finite-rate systems with radiation. *Intern. J. Heat and Mass Transfer*, 49(17-18), 3264-3283.
- Tsirlin, A. M., Kazakov, V., Mironova, V. A., Amelkin, S. A., 1998. Finite-time thermodynamics: conditions of minimal dissipation for thermodynamic process. *Physical Review E*, 58 (1), 215-223.
- Wierzbicki, M., 2009. *Optimization of SOFC based energy system using Aspen PlusTM*, Thesis (MsD) supervised by S. Sieniutycz (Faculty of Chemical and Process Engineering, Warsaw University of Technology) and J. Jewulski (Laboratory of Fuel Cells, Institute of Energetics, Warsaw).
- Zhao, Y., Ou, C., Chen, J., 2008. A new analytical approach to model and evaluate the performance of a class of irreversible fuel cells. *International Journal of Hydrogen Energy*, 2008 (33), 4161-4170.

AUTHORS BIOGRAPHY

Prof. **Stanislaw Sieniutycz** (1940), PhD; ScD, has been since 1983 a Professor Chemical Engineerig at the Warsaw Technological University, Warsaw Poland. He has received his MsD in Chemistry in 1962, PhD in Chemical Engineering in 1968, and ScD (habilitation) in Chemical Engineering in 1973, all from Warsaw TU. He is recognized for his applications of methods of analytical mechanics and optimal control theory in thermodynamics and engineering. Within this framework his research embraces a range of topics: efficiency of energy conversion, optimization of separation systems and chemical reactors, energy savings, conservation laws, wave systems, heat and mass transfer and variational principles in continua.

His work with graduates has resulted in several PhD theses: Discrete Maximum Principle of Pontryagin's type (Z. Szwasz 1979); Thermodynamic Liapounov Functions in Gas-Solid Systems (J. Komorowska-Kulik, 1979); Optimizations in Catalytic Cracking (A. Kebang 1980 and A. Dlugosz 1983); Modeling and Optimization in Separation of Coal Conversion Liquids (A. Dunalewicz 1989), Nonlinear Models of Mechanical Energy Production in Non-Ideal Generators Driven by Thermal or Solar Energy (M. Kubiak 2005 and P. Kuran 2006). **Prof. Sieniutycz** has contributed about 250 research papers in the field of chemical engineering and irreversible thermodynamics. He is also an author of several books: Optimization in Process Engineering (WNT Warsaw 1978 and 1991); Practice in Optimization Computations (with Z. Szwasz; WNT, Warsaw 1982); Conservation Laws in Variational Thermo-Hydrodynamics (Kluwer Academic, Dordrecht 1995), and, recently, with Prof. J. Jeżowski, Energy Optimization in Process Systems, (Elsevier, Oxford and Radarweg, 2009) He has been a visiting professor in a number of schools: University of Budapest (Physics), University of Bern (Physiology), University of Trondheim (Chemical Physics), San Diego SU (Mathematics), University of Delaware (Chemical Engineering), and (several times at) The University of Chicago (Chemistry).

OPTIMIZATION OF FLOWS AND ANALYSIS OF EQUILIBRIA IN TELECOMMUNICATION NETWORKS

Ciro D'Apice^(a), Rosanna Manzo^(b), Luigi Rarità^(c)

^{(a), (b), (c)}Department of Information Engineering and Applied Mathematics,
Via Ponte Don Melillo, 84084, Fisciano, Salerno, Italy

^(a)dapice@diima.unisa.it, ^(b)manzo@diima.unisa.it, ^(c)lrarita@unisa.it

ABSTRACT

The aim of this paper is the analysis of flows on data networks in order to improve traffic conditions. In particular, the attention is focused on optimization techniques and equilibrium solutions for a fluid dynamic model of telecommunication networks.

The optimization algorithm allows to redistribute the packets at nodes to avoid congestions. The performance analysis is made through a cost functional measuring the average velocity of packets and depending on priority and distribution coefficients at nodes.

The study of the equilibrium flows gives an understanding of the asymptotic solution on the whole network and permits the investigation of security issues when some nodes of the network fail.

Here, on the basis of analytical studies, simulative results are obtained. Simulations confirm that the optimization algorithm gives better performances with respect to other choices of the model parameters. For the equilibrium analysis, an iterative algorithm is studied in order to determine outflows equilibria as function of the input flows.

Keywords: conservation laws, simulation, optimization, equilibria.

1. INTRODUCTION

The aim of this work is to present some numerical results about the analysis of packets flows on data networks in order to understand typical phenomena, such as packets congestion. Starting from a fluid-dynamic model, introduced in D'Apice 2006 and refined in D'Apice 2008, we have tested some optimization algorithms (Marigo 2006) for the optimal choice of the model parameters and faced, from a numerical point of view, the problem of identifying equilibrium solutions (Marigo 2007).

As for the model of data networks, looking at an intermediate time scale, the average amount of packets is considered conserved, hence the density evolution for each line can be described by a conservation law. The dynamics at nodes, in which many lines intersect, is given by a routing algorithm, according to which packets are processed by arrival time (FIFO policy) and sent to the outgoing lines in order to maximize the flux on both incoming and outgoing lines.

To determine uniquely the evolution at nodes, some parameters (priority parameters and traffic distribution coefficients) have to be introduced. Such parameters have been considered in Marigo 2006 as controls and a functional, which measures the network efficiency in terms of the packets velocity, has been analysed. The optimization approach is of decentralized type, and consists in the following steps: first, the optimal parameters for simple networks, formed by a single node and every constant initial data, are evaluated; then, for a complex network, the (locally) optimal parameters at every node, updating the value of the parameters at every time instant, are used for testing the global performance on the network. Similar optimization approaches have already been considered for car traffic networks in Cascone 2007 and Cascone 2008.

For the first step, we focus the attention on a simple network characterized by two incoming lines, two outgoing lines (node of 2×2 type), a traffic distribution coefficient, α , and a priority parameter, p . Simulations prove that, setting the parameters α and p with the optimal ones, the traffic conditions at the node improve.

For the second step, we consider an Oriented Manhattan network, characterized by nodes of 2×2 type. Three different choices for the traffic distribution coefficients and priority parameters are analyzed: (locally) optimal, static random and fixed. The first choice is given by the optimal values. By static random parameters, we mean a random choice done at the beginning of the simulation and then kept constant. Finally, the fixed choice is represented by the simulation with fixed priority parameters and distribution coefficients. Simulations prove that the optimal algorithm, with respect to other traffic choices, gives the best performances as for the packets traffic evolutions.

The optimization analysis is completed by some numerical results on network equilibrium flows, that are reached in the asymptotic state (see Marigo 2007). Equilibria for data networks, which represent solutions constant in time on the whole network, can be used to face security issues in case of node failures.

The analysis of equilibria is carried out in the following way. First, we determine the possible types of equilibria at each node, depending on the combination of good and bad values at the incident lines. Then, we deduce the types for subnetworks, which are considered

the building blocks of the whole network. In this way, we understand which are all types of equilibria configuration on a generic network. From the knowledge of the type of equilibria configuration, we define a matrix equation that involves incoming and outgoing flows of the network and describes analytically the total asymptotic solution.

The paper is organized as follows. In Section 2, we introduce the model and give basic definitions and results for equilibrium solutions. Section 3 is about the optimization algorithm for a junction of 2×2 type. Section 4 considers equilibria configuration on an Oriented Manhattan Network. The paper ends with simulation results in Section 5.

2. A FLUID DYNAMIC MODEL FOR TELECOMMUNICATION NETWORKS

A network is a finite collection of transmission lines and nodes (or routers).

It is modeled by a finite set of intervals $I_i = [a_i, b_i] \subset R$, $i = 1, \dots, L$, $a_i < b_i$. We assume that the transmission lines are connected by some nodes. We identify each node J with $((i_1, \dots, i_m), (j_1, \dots, j_n))$, where the first m -tuple indicates the set of incoming lines and the second n -tuple the set of outgoing ones. Each transmission line is considered incoming at most for one node and outgoing at most for one node. Hence, the complete model is given by the couple (\mathbf{I}, \mathbf{J}) , where $\mathbf{I} = \{I_i : i = 1, \dots, L\}$ and \mathbf{J} are the collections of transmission lines and nodes, respectively. We set N to be the cardinality of \mathbf{J} .

We assume that each packet travels on the network with a fixed speed and an assigned final destination. Considering an intermediate time scale and assuming the conservation of packets, the load dynamics for a single line of the network is described by the conservation law (see D'Apice 2006):

$$\rho_t + f(\rho)_x = 0, \quad (1)$$

where $\rho = \rho(t, x) \in [0, \rho_{\max}]$, $(t, x) \in R^2$, is the density of packets, ρ_{\max} is the maximal density, $f(\rho) = v\rho$ is the flux with v the average velocity. In what follows, we suppose that:

(F) the flux f is a strictly concave function, with $f(0) = f(\rho_{\max}) = 0$.

Setting, for simplicity, $\rho_{\max} = 1$, a flux function ensuring (F) is:

$$f(\rho) = \rho(1 - \rho). \quad (2)$$

which has a unique maximum $\sigma = \frac{1}{2}$ over the interval $[0, \rho_{\max}]$.

In order to solve the dynamics at nodes, we follow the strategy proposed in D'Apice 2006 and consider the routing algorithm:

(RA) Packets are processed by arrival time and are sent to outgoing lines in order to maximize the flux on incoming and outgoing lines.

The main ingredient for finding an approximate solution of the Cauchy problem at a node with the wave-front tracking algorithm (see Garavello 2006) is given by the solution of a Riemann Problem (RP), which is a Cauchy problem with a constant initial datum on each incoming and outgoing line.

Using φ as index for the incoming lines and ψ for the outgoing ones, we introduce the following:

Definition. A Riemann Solver for a node J is a map RS that associates to Riemann data $\rho_0 = (\rho_{\varphi,0}, \rho_{\psi,0})$ at J a vector $\hat{\rho} = (\hat{\rho}_{\varphi}, \hat{\rho}_{\psi})$ so that the solution on an incoming line I_{φ} is given by the wave $(\rho_{\varphi,0}, \hat{\rho}_{\varphi})$ and on an outgoing one I_{ψ} by the wave $(\hat{\rho}_{\psi}, \rho_{\psi,0})$. The following conditions are required:

(C1) $RS(RS(\rho_0)) = RS(\rho_0)$.

(C2) On each incoming line, the wave $(\rho_{\varphi,0}, \hat{\rho}_{\varphi})$ has negative speed, while on each outgoing line $(\hat{\rho}_{\psi}, \rho_{\psi,0})$ has positive speed.

Definition. Let $\tau: [0,1] \rightarrow [0,1]$ be the map such that $f(\rho) = f(\tau(\rho))$ and $\tau(\rho) \neq \rho$ if $\rho \neq \sigma$.

From condition (C2), we have the following:

Proposition. Let RS be the Riemann Solver for a node. Assuming an initial data $\rho_0 = (\rho_{\varphi,0}, \rho_{\psi,0})$ and the flux function (2), then

$$\hat{\rho}_{\varphi} \in \begin{cases} \{\rho_{\varphi,0}\} \cup [\tau(\rho_{\varphi,0}), 1], & \text{if } 0 \leq \rho_{\varphi,0} < \frac{1}{2}, \\ \left[\frac{1}{2}, 1\right], & \text{if } \frac{1}{2} \leq \rho_{\varphi,0} \leq 1, \end{cases} \quad (3)$$

$\varphi = 1, \dots, m$, and

$$\hat{\rho}_\psi = \begin{cases} \left[0, \frac{1}{2}\right], & \text{if } 0 \leq \rho_{\psi,0} \leq \frac{1}{2}, \\ \{\rho_{\psi,0}\} \cup \left[0, \tau(\rho_{\psi,0})\right], & \text{if } \frac{1}{2} < \rho_{\psi,0} \leq 1, \end{cases} \quad (4)$$

$$\psi = m+1, \dots, m+n.$$

To describe the solution on each transmission line, it is enough to specify the flux values $\hat{\gamma}_\varphi = f(\hat{\rho}_\varphi)$ and $\hat{\gamma}_\psi = f(\hat{\rho}_\psi)$.

2.1. Riemann Solver according to rule (RA)

In order to have a unique solution of a RP at a node, based on rule (RA), some additional parameters, called, respectively, priority and traffic distribution parameters have to be defined. Precisely, we have m priority

parameters (p_1, p_2, \dots, p_m) , $\sum_{i=1}^m p_i = 1$, $p_i \in]0, 1[$, for

the incoming lines and n traffic distribution parameters

$(\alpha_1, \alpha_2, \dots, \alpha_n)$, $\sum_{i=1}^n \alpha_i = 1$, $\alpha_i \in]0, 1[$, for the outgoing

ones.

Assume $(\rho_{\varphi,0}, \rho_{\psi,0})$ the initial data and use the notation $\gamma_i = f(\rho_{i,0})$, $i = a, b, c, d$.

Let the maximum fluxes, γ_φ^{\max} and γ_ψ^{\max} , and the maximal through flux Γ at J be defined as follows:

$$\gamma_\varphi^{\max} = \begin{cases} \gamma_\varphi, & \text{if } \rho_{\varphi,0} \in \left[0, \frac{1}{2}\right], \\ f(\sigma), & \text{if } \rho_{\varphi,0} \in \left[\frac{1}{2}, 1\right], \end{cases} \quad \varphi = 1, \dots, m, \quad (5)$$

$$\gamma_\psi^{\max} = \begin{cases} f(\sigma), & \text{if } \rho_{\psi,0} \in \left[0, \frac{1}{2}\right], \\ \gamma_\psi, & \text{if } \rho_{\psi,0} \in \left[\frac{1}{2}, 1\right], \end{cases} \quad \psi = m+1, \dots, m+n. \quad (6)$$

$$\Gamma = \min \{ \Gamma_{in}, \Gamma_{out} \}, \quad (7)$$

where:

$$\Gamma_{in} = \sum_{\varphi=1}^m \gamma_\varphi^{\max}, \quad \Gamma_{out} = \sum_{\psi=m+1}^{m+n} \gamma_\psi^{\max}. \quad (8)$$

We give the solution of a RP at a node with two incoming lines, a and b , and two outgoing lines, c and d . Introduce the conditions:

$$(A1) \quad \alpha\Gamma < \gamma_c^{\max};$$

$$(A2) \quad (1-\alpha)\Gamma < \gamma_d^{\max}.$$

In the case $\Gamma_{in} = \Gamma$, we get that:

- if (A1) and (A2) are both satisfied, then $(\hat{\gamma}_a, \hat{\gamma}_b, \hat{\gamma}_c, \hat{\gamma}_d) = (\gamma_a^{\max}, \gamma_b^{\max}, \alpha\Gamma, (1-\alpha)\Gamma)$;
- if (A1) does not hold and (A2) is satisfied, then $(\hat{\gamma}_a, \hat{\gamma}_b, \hat{\gamma}_c, \hat{\gamma}_d) = (\gamma_a^{\max}, \gamma_b^{\max}, \gamma_c^{\max}, \Gamma - \gamma_c^{\max})$;
- if (A1) holds and (A2) does not hold, then $(\hat{\gamma}_a, \hat{\gamma}_b, \hat{\gamma}_c, \hat{\gamma}_d) = (\gamma_a^{\max}, \gamma_b^{\max}, \Gamma - \gamma_c^{\max}, \Gamma - \gamma_d^{\max})$.

In the case $\Gamma_{out} = \Gamma$, the solution of the RP depends on the priority parameter p (for details, see D'Apice 2006).

2.2. Equilibrium solutions

Now, we give a classification of the solutions to the RP at a node.

Definition. A component of the solution at a node, $\hat{\rho}_\varphi$, $\varphi = 1, \dots, m$, is:

- *bad* if $\hat{\rho}_\varphi \in \left[0, \frac{1}{2}\right[$;
- *good* if $\hat{\rho}_\varphi \in \left[\frac{1}{2}, 1\right]$;

a component of the solution, $\hat{\rho}_\psi$, $\psi = m+1, \dots, m+n$, is:

- *bad* if $\hat{\rho}_\psi \in \left[\frac{1}{2}, 1\right]$;
- *good* if $\hat{\rho}_\psi \in \left[0, \frac{1}{2}\right]$.

We are interested in equilibrium solutions.

Definition. An equilibrium is a solution $\rho(t, x) = (\rho_1, \dots, \rho_L)$, which is constant in time. We also assume that $\rho(t, \cdot)$ is BV, thus we can define, for every $i = 1, \dots, L$, the following:

$$\rho_i^- = \lim_{x \rightarrow a_i} \rho(t, x), \quad \rho_i^+ = \lim_{x \rightarrow b_i} \rho(t, x). \quad (9)$$

Since ρ is a solution, then

$$\sum_{\varphi=1}^m f(\rho_{j_\varphi}) = \sum_{\psi=m+1}^{m+n} f(\rho_{j_\psi}), \quad (10)$$

is satisfied at each node $J_j \in \mathcal{J}$, $j = 1, \dots, N$. We distinguish two cases:

- (i) there exists $i = 1, \dots, L$, such that $\rho_i^- \neq \rho_i^+$. In this case, $\rho_i^+ = \tau(\rho_i^-)$ and

the fluxes $\gamma_i = f(\rho_i^\pm)$, are anyhow constant in time and along the whole line I_i ;

- (ii) for all $i=1, \dots, L$, $\rho_i^+ = \rho_i^-$ and we call this value ρ_i .

Definition. Let $\rho = (\rho_1, \dots, \rho_L)$ be an equilibrium for the network (\mathbf{I}, \mathbf{J}) , satisfying (ii). We say that ρ_1, \dots, ρ_L are the values of the equilibrium. Moreover, if ρ_i is of type τ_i , with $\tau_i \in \{\text{bad}, \text{good}\}$, then we say that $T = (\tau_1, \dots, \tau_L)$ is the equilibrium type.

2.3. Examples

For simplicity, we consider a telecommunication network consisting of only one node, o , two sources, $\{1, 2\}$, two destinations, $\{3, 4\}$ and four lines, $\{a, b, c, d\}$, as in Figure 1.

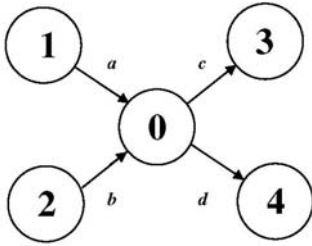


Figure 1: A node of type 2×2

Remark. The RS for the node o depends on one priority coefficient, α , and one distribution parameter, p .

Fix the flux function (2), the distribution coefficient $\alpha = 0.6$, the priority parameter $p = 0.2$, and the initial data $\rho_0 = (\rho_{a,0}, \rho_{b,0}, \rho_{c,0}, \rho_{d,0}) = (0.3, 0.2, 0.25, 0.55)$.

From (8) we have that:

$$\begin{aligned} \Gamma_{in} &= \gamma_a^{\max} + \gamma_b^{\max} = 0.37; \\ \Gamma_{out} &= \gamma_c^{\max} + \gamma_d^{\max} = 0.4975. \end{aligned} \quad (11)$$

As $\Gamma = \Gamma_{in} < \Gamma_{out}$, the fluxes, solution to the RP, at J are given by:

$$\begin{aligned} \hat{\gamma} &= (\gamma_a^{\max}, \gamma_b^{\max}, \alpha\Gamma, (1-\alpha)\Gamma) = \\ &= (0.21, 0.16, 0.222, 0.148). \end{aligned} \quad (12)$$

Notice that, in this case, the solution does not depend on the priority parameter p . The corresponding densities are:

$$\hat{\rho} = (0.3, 0.2, 0.332668, 0.180626). \quad (13)$$

Since $\hat{\rho}_\varphi = \rho_{\varphi,0} < \frac{1}{2}$, $\varphi = a, b$, $\hat{\rho}_\varphi$, $\varphi = a, b$, is a bad data; $\hat{\rho}_\psi < \frac{1}{2}$, $\psi = c, d$, thus $\hat{\rho}_\psi$, $\psi = c, d$, is a good data.

Assume now $\alpha = 0.6$, $p = 0.4$ and an initial data $\rho_0 = (0.55, 0.75, 0.65, 0.85)$.

We have that $\Gamma = \Gamma_{out} < \Gamma_{in}$, thus the fluxes, solution to the RP, at J are given by:

$$\begin{aligned} \hat{\gamma} &= (p\Gamma, (1-p)\Gamma, \gamma_c^{\max}, \gamma_d^{\max}) = \\ &= (0.142, 0.213, 0.2275, 0.1275). \end{aligned} \quad (14)$$

Observe that the solution does not depend on the distribution coefficient α . The corresponding densities are:

$$\hat{\rho} = (0.828634, 0.692324, 0.65, 0.85). \quad (15)$$

Since $\hat{\rho}_\varphi > \frac{1}{2}$, $\varphi = a, b$, we get that $\hat{\rho}_\varphi$, $\varphi = a, b$, is a good data; moreover, $\hat{\rho}_\psi = \rho_{\psi,0} > \frac{1}{2}$, $\psi = c, d$, thus $\hat{\rho}_\psi$, $\psi = c, d$, is a bad data.

3. OPTIMIZATION OF DATA NETWORKS

Consider a junction J with m incoming lines and n outgoing lines (junction of type $m \times n$). For an initial data $\rho_0 = (\rho_{\varphi,0}, \rho_{\psi,0})$, we define the cost functional $W(t)$ as:

$$W(t) = \int_{I_\varphi} v(\rho_\varphi(t, x)) dx + \int_{I_\psi} v(\rho_\psi(t, x)) dx, \quad (16)$$

$\varphi = 1, \dots, m$, $\psi = m+1, \dots, m+n$.

Such functional indicates the average velocity of packets that, from an incoming line I_φ , $\varphi = 1, \dots, m$, are directed to the outgoing line I_ψ , $\psi = n+1, \dots, n+m$. For a fixed time horizon $[0, T]$, we want to maximize $\int_0^T W(t) dt$ by a suitable choice of distribution coefficients $\alpha_\psi \in]0, 1[$, $\psi = m+1, \dots, m+n$ if $\Gamma_{in} = \Gamma$ or priority parameters $p_\varphi \in]0, 1[$, $\varphi = 1, \dots, m$ if $\Gamma_{out} = \Gamma$.

3.1. The case $m = 2$ and $n = 2$

Fix the flux function (2) and focus on a network with a junction of 2×2 type. For T sufficiently big, the cost functional $W(T)$ can be written as:

$$W(T) = \sum_{i \in \{a,b,c,d\}} v(\hat{\rho}_i) = 2 - \frac{1}{2} \sum_{i \in \{a,b,c,d\}} s_i \sqrt{1 - 4\hat{\gamma}_i}, \quad (17)$$

where:

for $\varphi = a, b$:

$$s_\varphi = -1 \quad \text{if} \quad \rho_{\varphi,0} \leq \frac{1}{2} \quad \text{and} \quad \Gamma = \Gamma_{in}, \quad \text{or} \quad \rho_{\varphi,0} \leq \frac{1}{2},$$

$$p_\varphi \Gamma = \gamma_\varphi^{\max} \quad \text{and} \quad \Gamma = \Gamma_{out};$$

$$s_\varphi = +1 \quad \text{if} \quad \rho_{\varphi,0} > \frac{1}{2}, \quad \text{or} \quad \rho_{\varphi,0} \leq \frac{1}{2}, \quad p_\varphi \Gamma < \gamma_\varphi^{\max} \quad \text{and}$$

$$\Gamma = \Gamma_{out};$$

for $\psi = c, d$:

$$s_\psi = +1 \quad \text{if} \quad \rho_{\psi,0} \geq \frac{1}{2} \quad \text{and} \quad \Gamma = \Gamma_{out}, \quad \text{or} \quad \rho_{\psi,0} \geq \frac{1}{2},$$

$$\alpha_\psi \Gamma = \gamma_\psi^{\max} \quad \text{and} \quad \Gamma = \Gamma_{in};$$

$$s_\psi = -1 \quad \text{if} \quad \rho_{\psi,0} < \frac{1}{2}, \quad \text{or} \quad \rho_{\psi,0} \geq \frac{1}{2}, \quad \alpha_\psi \Gamma < \gamma_\psi^{\max} \quad \text{and}$$

$$\Gamma = \Gamma_{in}, \quad \text{with:}$$

$$p_\varphi = \begin{cases} p, & \text{if } \varphi = a, \\ 1-p, & \text{if } \varphi = b, \end{cases} \quad \alpha_\psi = \begin{cases} \alpha, & \text{if } \psi = c, \\ 1-\alpha, & \text{if } \psi = d. \end{cases}$$

We report results for the optimal choice of α . Regarding the choice of the optimal priority parameter, refer to Cascone 2007 and Marigo 2006.

If $\Gamma = \Gamma_{in}$, $\hat{\gamma}_\varphi = \gamma_\varphi^{\max}$, $\varphi = a, b$, hence maximizing (17) is equivalent to maximize

$$\hat{W}(T) = -s_c \sqrt{1 - 4\hat{\gamma}_c} - s_d \sqrt{1 - 4\hat{\gamma}_d}. \quad (18)$$

The optimal choice of α , α_{opt} , for different choice of s_c and s_d is as follows:

- if $\beta^- \leq 1 \leq \beta^+$ and $s_c = s_d = -1$ or $s_c = -1 = -s_d$ or $s_c = +1 = -s_d$, $\alpha_{opt} = 0.5$;
- if $\beta^- \leq \beta^+ \leq 1$ and $s_c = s_d = -1$ or $s_c = +1 = -s_d$, $\alpha_{opt} \in \left[0, \frac{1}{1 + \beta^+}\right]$;
- if $\beta^- \leq \beta^+ \leq 1$ and $s_c = -1 = -s_d$, α_{opt} does not exist and one can choose $\alpha_{opt} = \frac{1}{1 + \beta^+} + \varepsilon$;
- if $1 \leq \beta^- \leq \beta^+$ and $s_c = s_d = -1$ or $s_c = -1 = -s_d$, $\alpha_{opt} \in \left[\frac{1}{1 + \beta^-}, 1\right]$;

- if $1 \leq \beta^- \leq \beta^+$ and $s_c = +1 = -s_d$, α_{opt} does not exist and one can choose

$$\alpha_{opt} = \frac{1}{1 + \beta^-} - \varepsilon,$$

$$\text{where } \beta^- = \frac{\Gamma - \gamma_c^{\max}}{\gamma_c^{\max}} \quad \text{and} \quad \beta^+ = \frac{\gamma_d^{\max}}{\Gamma - \gamma_d^{\max}}.$$

4. EQUILIBRIA IN MANHATTAN TYPE NETWORKS

Let us focus on oriented square networks with $s \times t$ nodes of 2×2 type as in the Figure 2, which we call ‘‘Oriented Manhattan’’ (OM briefly).

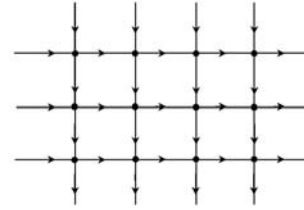


Figure 2: topology of an OM network

In Marigo 2007, we got the following.

Proposition. Consider a network (\mathbf{I}, \mathbf{J}) , where $N = s \times t$ is the cardinality of \mathbf{J} , while $L = 4 \times s \times t$ is the cardinality of \mathbf{I} . The set of equilibrium values is a $L - N = 3 \times s \times t$ dimensional subspace of $\mathbb{R}^{4 \times s \times t}$.

According to such proposition, at the equilibrium, on each line $I \in \mathbf{I}$, there is a flow among two adjacent nodes which is constant in time and along the line I (L variables) with the constraint (10) at each node J (for a total of N constraints). This result is based only on the topological property of the network.

We consider an equilibrium at one node of a square network, and describe componentwise its type *good* or *bad*.

The possible equilibrium types at one node are the followings (we use the short notations *b* for bad and *g* for good):

$$I : ((b, b), (b, b));$$

$$II0 : ((b, b), (g, g)); \quad III1.1 : ((b, b), (b, g));$$

$$III1.2 : ((b, b), (g, b)); \quad III0 : ((g, g), (b, b));$$

$$III1.1 : ((b, g), (b, b)); \quad III1.2 : ((g, b), (b, b)).$$

Definition. The set of all possible equilibria types at a node is $\mathbf{M} = \{I, II0, III1.1, III1.2, III0, III1.1, III1.2\}$. The set of all possible equilibria types for the whole network is $\hat{\mathbf{N}} = \{\delta_{ij} \in \mathbf{M}, i = 1, \dots, s, j = 1, \dots, t\} = \mathbf{M}^{s \times t}$.

Clearly, the cardinality of $\hat{\mathbf{N}}$ is $M^{s \times t}$. However, not all the elements of $\hat{\mathbf{N}}$ may arise. Indeed, two kinds of equilibria over the whole network

may be considered. In case (ii), the following compatibility rule must be satisfied:

(H) if a line I_i is incoming for some node J_1 and outgoing for some other node J_2 , then whenever $\hat{\rho}_i$ is of type bad for J_1 , then it must be of type good for J_2 and viceversa.

Rule (H) gives rise to a compatibility relation among equilibria at adjacent nodes, which determines the subset $\mathbf{N} \subset \widehat{\mathbf{N}}$ of admissible equilibrium states for the whole network. This fact does not hold for case (i). Consider now the equilibria (ii) and give a characterization of \mathbf{N} . Indicate by a and b the incoming lines while by c and d the outgoing lines.

Then, the following holds:

Proposition. Consider a node J of 2×2 type and assume that $(\rho_{a,0}, \rho_{b,0}, \rho_{c,0}, \rho_{d,0})$ is an equilibrium.

If J is of type $II0$, $\gamma_\varphi = \gamma_\varphi^{\max}$, $\varphi = a, b$, while $\gamma_c = \alpha \Gamma_{in}$ and $\gamma_d = (1 - \alpha) \Gamma_{in}$.

For other types of equilibria, similar proposition hold (see Marigo 2007).

Remark. Note that if $\rho_0 = (\rho_{a,0}, \rho_{b,0}, \rho_{c,0}, \rho_{d,0})$ is an equilibrium, then $\hat{\rho} = (\hat{\rho}_a, \hat{\rho}_b, \hat{\rho}_c, \hat{\rho}_d) = RS(\rho_0) = \rho_0$ and $\hat{\gamma}_i = \gamma_i = f(\rho_{i,0})$, $i = a, b, c, d$.

A $s \times t$ network is described by $4 \times s \times t$ parameters $in_{ij}^h, in_{ij}^v, out_{ij}^h, out_{ij}^v$, $i = 1, \dots, s$, $j = 1, \dots, t$, together with $4s \times t - s - t - \omega$ constraints, where $\omega \leq \min(s, t)$ is the number of nodes of type I , while $in_{ij}^h, in_{ij}^v, out_{ij}^h, out_{ij}^v$ are, respectively: incoming flows for horizontal and vertical lines; outgoing flows for horizontal and vertical lines.

In what follows, we consider only networks of type $II0$. For such a network, it is shown that a matrix equation rules the behaviour of the outgoing flows as function of incoming flows (for the proof see Marigo 2007):

$$AOUT + OUTB + in = 0, \quad (19)$$

where we have: $OUT = \left(out_{ij}^h \right)_{ij}$, $A = \alpha(-I(s) + L(s))$, $B = (-I(t) + L(t))^T$, with $I(n)$ the identity matrix of order n , $L(n)$ the sub-diagonal matrix of order n , in the inflow vector:

$$L = \begin{bmatrix} 0 & & & & & & \\ 1 & 0 & & & & & \\ & & \ddots & & & & \\ & & & \ddots & & & \\ & & & & 1 & 0 & \\ & & & & & & 0 \end{bmatrix}, in = \begin{bmatrix} x_1^h + x_1^v & x_2^h & x_3^h & x_4^h & x_5^h & \dots & x_t^h \\ x_2^v & & & & & & 0 \\ x_3^v & & & & & & 0 \\ x_4^v & & & \circ & & & 0 \\ \vdots & & & & & & \vdots \\ x_s^v & 0 & 0 & 0 & 0 & \dots & 0 \end{bmatrix}$$

and x_i^h , $i = 1, \dots, t$ and x_i^v , $i = 1, \dots, s$ are the incoming flows for horizontal and vertical line.

The equation (19) is a Sylvester matrix equation, and its solution is found recursively as:

$$OUT = \frac{1}{2} C_\infty, \quad (20)$$

where:

$$A_0 = A, B_0 = B, C_0 = in, \quad (21)$$

$$C_{k+1} = \frac{1}{2}(C_k + A_k^{-1} C_k B_k^{-1}).$$

Notice that (20) gives the analytical solution on the whole network in terms of outgoing flows, under the assumption of equal distribution coefficients for all nodes of the network.

5. SIMULATIONS

In this section, we present some simulation results in order to test the optimization algorithms either for single nodes or complex networks. The aim is to verify the correctness of the analytical results and to analyse the effects of different choices of distribution and priority parameters, applied locally at each junction, on the global performances of the network.

5.1. Single junctions

Consider a telecommunication network with only one node of 2×2 type. We compare the different behaviours of the cost functional (16) in such situations: use of the optimization algorithm (*optimal case*); adoption of fixed priority parameters and distribution coefficients (*fixed case*), namely parameters chosen by the user.

The evolution of packets traffic is simulated in a time interval $[0, T]$, where $T = 10$ min for the flux function (2). We assume that, at the starting instant of simulation ($t = 0$), the initial datum is $\rho_0 = (0.4, 0.35, 0.3, 0.2)$. Moreover, for lines a, b, c and d , we choose the following Dirichlet boundary data: $\rho_{a,b} = 0.4$, $\rho_{b,b} = 0.35$, $\rho_{c,b} = \rho_{d,b} = 0$.

Notice that the initial conditions and boundary data are such that the dynamics of the network is ruled only by the distribution coefficient α , whose optimal value is 0.5.

Figures 3 and 4 show that, setting the network with the optimal value α , traffic conditions are improved

with respect to fixed cases: the optimal cost functional is higher with respect to others. In particular, Figure 3 depicts the temporal behaviour of W in $[0, T]$, while Figure 4 represents a zoom of the cost functional behaviour. The latter Figure shows clearly that, in some contexts, although the asymptotic state is not reached (the final T), the optimization algorithm gives better performances with respect to other simulation cases.

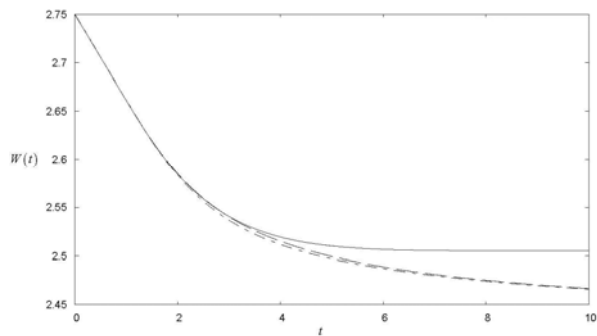


Figure 3: behaviour of $W(t)$ for optimal choice of α (solid line), $\alpha = 0.25$ (dashed line) and $\alpha = 0.65$ (dot dashed line)

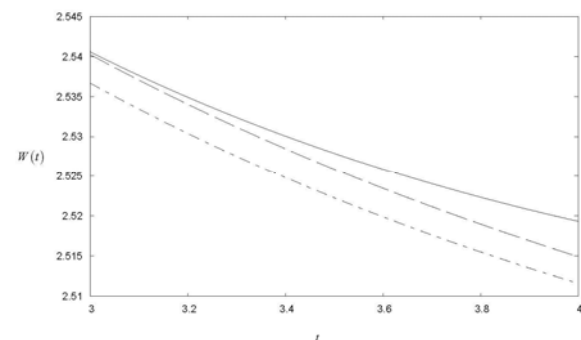


Figure 4: zoom of $W(t)$ for optimal choice of α (solid line), $\alpha = 0.25$ (dashed line) and $\alpha = 0.65$ (dot dashed line)

5.2. Complex networks

Here, we describe the simulation results for a complex OM network.

The network, represented in Figure 5, consists in 24 lines, divided into two subsets, $L_1 = \{a_2, a_3, d_1, d_2, d_3, c_2, c_3, f_1, f_2, f_3, e_2, e_3\}$, and $L_2 = \{a_1, a_4, b_1, b_2, b_3, c_1, c_4, e_1, e_4, g_1, g_2, g_3\}$ that are respectively, the set of inner and external lines. All nodes are of 2×2 type and are labelled by the couple (i, j) , where i and j are, respectively, the row and the column indices.

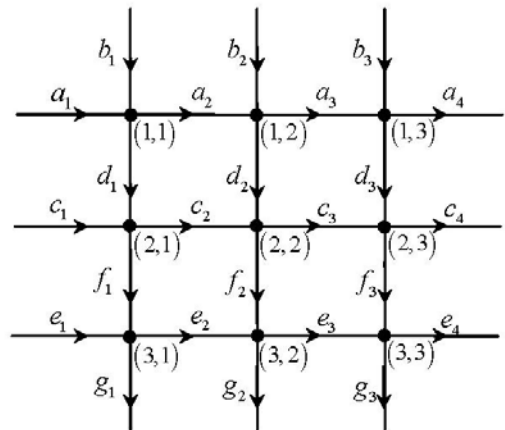


Figure 5: a complex network of OM type

The evolution of traffic flows is simulated in a time interval $[0, T]$, where $T = 30$ min. Initial conditions for all lines and boundary data are in the following table:

Table 1: Initial conditions and boundary data for all lines

Line	$\rho_{i,0}$	$\rho_{i,b}$	Line	$\rho_{i,0}$	$\rho_{i,b}$
a_1	0.2	0.3	d_2	0.2	/
a_2	0.3	/	d_3	0.25	/
a_3	0.2	/	e_1	0.2	0.35
a_4	0.4	0.3	e_2	0.2	/
b_1	0.1	0.3	e_3	0.1	/
b_2	0.15	0.25	e_4	0.3	0.15
b_3	0.25	0.35	f_1	0.3	/
c_1	0.3	0.4	f_2	0.2	/
c_2	0.25	/	f_3	0.1	/
c_3	0.3	/	g_1	0.4	0.25
c_4	0.15	0.1	g_2	0.2	0.1
d_1	0.3	/	g_3	0.25	0.15

We analyze different type of simulation cases: priority parameters and distribution coefficients, that optimize the cost functional (*optimal case*) in each node; fixed parameters (*fixed case*), which means that networks parameters are fixed for each node by the user at the beginning of the simulation process (in what follows, priority parameters are chosen equal to 0.3, while distribution coefficients are fixed at 0.2); random parameters (*random case*), where the parameters are randomly chosen when the simulation starts.

The aim is to understand the effects of the optimal algorithm of local type on the whole network. Figure 6 shows the temporal behaviour of the cost functional $W(t)$ for various choices of parameters at nodes. The optimal cost functional is higher than other simulation cases, hence the principal aim is achieved. Moreover,

when simulation begins, W with optimal value parameters is lower than the fixed case. In the steady state, instead, the optimal configuration is the highest (Figure 7).

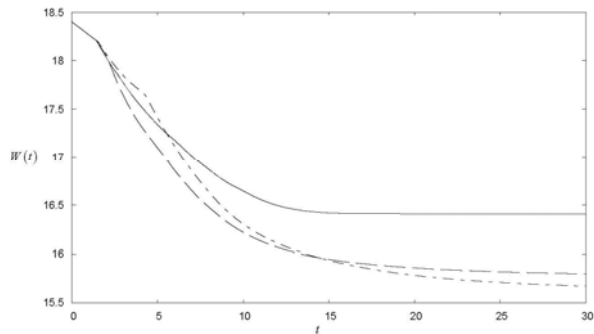


Figure 6: $W(t)$ for optimal choice of network parameters (solid line), random parameters (dashed line) and fixed parameters (dot dashed line)

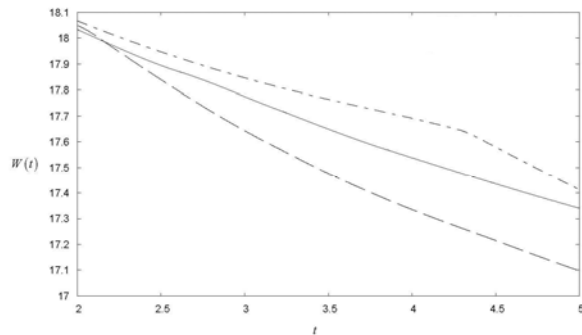


Figure 7: zoom of $W(t)$ in the time interval $[2, 5]$; optimal choice of network parameters (solid line), random parameters (dashed line) and fixed parameters (dot dashed line)

The optimization procedure can be also applied to telecommunication networks with hundreds of nodes and arcs and the computational times are not very expensive. This is due to the few parameters, which characterize fluid dynamic models.

5.3. Equilibria

In this section, we consider the analysis of equilibria of type IIO , $((b, b), (g, g))$.

We first analyze a node of 2×2 type. An equilibrium of type IIO is given by $\rho_0 = (0.15, 0.3, 0.45, 0.1)$ with $\alpha = 0.733$. In this case, $\Gamma = \Gamma_{in} < \Gamma_{out}$, $\hat{\rho} = \rho_0$ and $\hat{\gamma}_i = \gamma_i = f(\rho_{i,0})$. Observe that $\hat{\rho}_i < \sigma$, $i = a, b, c, d$, hence the equilibrium is (b, b, g, g) .

Consider now an OM network (Figure 5) and apply the iterative algorithm in order to find the equilibrium outflows. Assume that, for each node, the distribution coefficient is $\alpha = 0.3$, and the inflow matrix, in , is the following:

$$in = \begin{pmatrix} x_1^h + x_1^v & x_2^h & x_3^h \\ x_2^v & 0 & 0 \\ x_3^v & 0 & 0 \end{pmatrix} = \begin{pmatrix} 0.6 & 0.25 & 0.35 \\ 0.4 & 0 & 0 \\ 0.35 & 0 & 0 \end{pmatrix},$$

where x_i^h , $i = 1, 2, 3$, are the inflows for lines, respectively, b_1, b_2, b_3 , while x_i^v , $i = 1, 2, 3$, are the inflows for a_1, c_1, e_1 .

Using equation (19), we obtain the following outflows:

Table 2: equilibrium outflows values

Line	outflow
a_2	0.461538
a_3	0.547337
a_4	0.690259
c_2	0.414201
c_3	0.444925
c_4	0.501541
e_2	0.364816
e_3	0.383302
e_4	0.4109588

6. CONCLUSIONS

In this paper, the analysis of packets flows on telecommunication networks has been considered. Some optimization procedures for parameters at junctions and equilibrium solutions have been investigated.

Simulations proved that the optimization algorithm for packets velocities allows to redistribute data flows so as to avoid congestions. The algorithm was tested either on a single junction or on a network with more nodes, concluding that real benefits on traffic performances are possible.

The study of the equilibrium flows, through a matrix equation, gave the possibility of knowing asymptotic solutions on a test complex network. Such results can be used to face the security issues when some nodes of the network fail.

REFERENCES

- Cascone A., D'Apice, C., Piccoli, B., and Rarità, L. Optimization of traffic on road networks. *Mathematical Models and Methods in Applied Sciences*, 17 (10), 1587–1617.
- Cascone A., D'Apice, C., Piccoli, B., and Rarità, L. Circulation of traffic in congested urban areas. *Communication in Mathematical Sciences*, 6 (3), 765–784.
- D'Apice, C., Manzo, R., and Piccoli, B., 2006. Packets flow on telecommunication networks. *SIAM Journal on Mathematical Analysis*, 38, 717–740.
- D'Apice, C., Manzo, R., and Piccoli, B., 2008. A Fluid Dynamic Model for Telecommunication Networks

with Sources and Destination. *SIAM Journal on Applied Mathematics*, 68 (4), 981–1003.

Marigo, A., 2006. Optimal Traffic Distribution and Priority Coefficients for Telecommunication Networks. *Networks and Heterogeneous Media*, 1 (2), 315–336.

Marigo, A., 2007. Equilibria for Data Networks. *Networks and Heterogeneous Media*, 2 (3), 497–528.

AUTHORS BIOGRAPHY

CIRO D'APICE was born in Castellammare di Stabia, Italy and obtained PhD degrees in Mathematics in 1996. He is associate professor at the Department of Information Engineering and Applied Mathematics of the University of Salerno. He is author of approximately 130 publications, with more than 70 journal papers about homogenization and optimal control; conservation laws models for vehicular traffic, telecommunications and supply chains; spatial behaviour for dynamic problems; queueing systems and networks.

His e-mail address is dapice@diima.unisa.it.

ROSANNA MANZO was born in Polla, Salerno, Italy. She graduated in Mathematics in 1996. She is a 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 and supply networks, optimal control, queueing theory, self – similar processes, computer aided learning. She is author of about 30 papers appeared on international journals and many publications on proceedings.

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 with a thesis about control problems for flows on networks. He is actually a research assistant at the University of Salerno. His scientific interests are about numerical schemes and optimization techniques for fluid – dynamic models, and queueing theory.

His e-mail address is lrarita@unisa.it.

This work is partially supported by MIUR-FIRB Integrated System for Emergency (InSyEme) project under the grant RBIP063BPH.

PROGRAMMING LANGUAGES AS TOOLS FOR STRUCTURED DESCRIPTION AND SIMULATION OF INFORMATION SYSTEMS

Eugene Kindler^(a), Ivan Křivý^(b), Cyril Klimes^(c)

Department of Informatics and Computers
University of Ostrava, Faculty of Science
CZ – 701 03 Ostrava, Dvorakova 7,
Czech Republic

^(a) ekindler@centrum.cz, ^(b) Ivan.Krivy@osu.cz, ^(c) Cyril.Klimes@osu.cz

ABSTRACT

The paper is oriented to general technique of the synthesis of two levels existing in information systems and in larger ones encompassing them. The technique is related to formal descriptions and to simulation of such systems. One level consists in abstract information processing and the other in existing in time. Expected changes of such systems may concern only one of the levels. The economy and ergonomics of the description and simulation is supported by model nesting, namely that the information processing model is nested in the simulation model of the physical system. The optimal way is applying programming languages that are object-oriented, block-oriented and process-oriented.

Keywords: object-oriented programming, nested models, information systems, simulation

1. INTRODUCTION – FORMAL APPARATUS NEED

1.1. Mathematics and Simulation Languages

If a computer model of a certain object is demanded, it must be exactly specified. Thus the concerned object Ω is analyzed so that a system S is “defined on it” and S should be transformed into model M able of running at a computer. The step from Ω to S is commonly called system analysis, while the second step is a certain type of programming. S should figure in exact communication

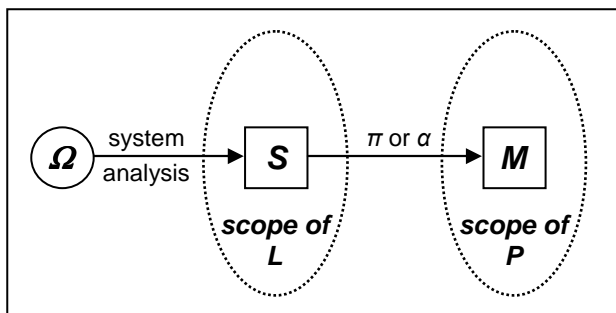


Figure 1: Basic Scheme

among humans and an ideal exists to describe S as exactly as it exists in mathematics. Nevertheless, if such a desire is realized, then the programming, i.e. the transformation from S to M , could be an object of an automatic translation α from the formal apparatus L (designed for describing S) into programming language P (designed for describing M – see Fig. 1 where the step of programming is represented by π).

It is evident that the (interested part of) human society desires language L to be as wide as possible and thus to become a universal communication tool for describing as many systems like P as possible. Related to that desire, L is often called specification language. Another desire is to reduce programming as much as possible, even in case it is replaced by automated process α . In other words, the desire is to draw language L to language P as near as possible. The ideal state is that programming and/or process α are totally reduced, i.e. that the specification language L is equal to the programming one P .

The first approach to realize something like that desire existed in simulation of systems that could be described with use of ordinary differential equations. However, the main stream in that development did not tend to reducing process α , managing the users to express the system analysis results as idealized analogue computers – e.g. language-standard CSSL (Strauss et al. 1967) or ACSL (Mitchel and Gauthier 1976) – there were many good languages based almost directly on the ordinary differential equations – e.g. DARE (Korn 1973).

While in the domain of the continuous systems standard mathematical languages appeared to be well applicable as a specification and programming language, it totally failed in the domain of discrete and combined systems. The central problem consisted in scheduling events called from rather different sources (elements of the concerned systems) in time instants that could not be globally predicted (even in a statistical sense). In practice, with a few of exceptions of discrete systems of “Mealy’s automata-like type” (like computers viewed at the level of logical gates or of microprogramming), the systems with discrete events are composed of rather great

number of elements that “live” in the same Newtonian time, during which each of them sometimes waits, sometimes modifies its own state, sometimes modifies the states of other elements and sometimes emits signals to them to do something (including an interruption of their waiting or – on the contrary – a prolongation of their waiting).

The elements of such a system can be viewed as instances of “classes” of similar elements. The instances of the same class have similar properties (values, “attributes”) and may behave (“live”, active figure in the system) according to similar “life rules” that can be formulated with help of standard algorithmic tools known in algorithmic programming (assignments, branchings, cycles, procedure calls,...). The similarity of the properties (attributes) consists in their common naming independently of difference among the instances of the class, and in some other common properties. Frequently the “contents” of the instances of the same attribute of different instances can be measured in the same manner, which is reflected by the same type (in the sense of programming) of these instances. It is not excluded that a system is composed of a great number of elements, which are instances of a rather small number of classes. Even in that case, common mathematics cannot help with its formal languages, because at the same time two instances of the same class generally behave according to different rules of the set of their common life rules. Moreover, an instance of a class can enter a system during its existence and even during operating or waiting phase of other instances of (the same or different) class.

Only elements of conventional algorithmic languages offer to contribute in the development of an exact tool that could become a specification language and possibly a programming language, too. But the only elements of the algorithmic languages do not suffice, as they are not able to reflect the parallelism of the active and interactive “lives” of different elements. The solving of the problem was started by the author of discrete event simulation language GPSS (Gordon 1960) by including “scheduling statements” into the life rules expressed as algorithms in class formulations. Thus a development of so called process oriented simulation languages started and was concluded by a discrete event simulation language SIMULA I (Dahl and Nygaard 1966). Note that in the title of the users’ manual (Dahl and Nygaard 1965), this language is presented not only as a programming one but – explicitly – as “language for *description* of discrete event systems”. So the simulation languages like SIMULA became non-traditional tools of mathematics.

It is suitable to know that a programming language can be considered as an abstract mathematical tool for system description only when it is isolated from hardware elements hidden under language texts and following from them in the compiled model. For example, a language cannot be considered specification one (exact tool for system description, etc.) if it admits using functions like *address(X)* for internal address of object *X* or/and

object(n) for pointing to an object that uses internal address *n* for its data. Such a language overpasses the description of the modeled (simulated) system and may “spoil” such a description by expressing something concerning the model of the described system; it is evident that in principle the detailed properties of the model can be essentially independent of the modeled system.

1.2. Mathematics and Object-Oriented Programming Tools

In order to make the language used for the description of a system as near as possible to the language for the description of the corresponding model, object-oriented programming (further OOP) arose. Nowadays, it is not frequently known that OOP arose as a reaction to “software crisis” in simulation, which demanded enormous number of new languages for the description of systems that tend to be more and more complex and to pass over boundaries of a certain conventional domain – the described and modeled systems reflected properties and problems formerly observed, processed and elaborated by different branches of science or technology. The first complete description of the OOP paradigm was presented by Dahl and Nygaard (1968) just at a conference on simulation programming languages, the aim of which was – not according to an official program but more or less subconsciously – to unify the communication on systems and consequently the description rules for them (Buxton, 1968). The next development led to a final definition of the proposed language (Dahl, Myhrhaug, and Nygaard, 1968) that was called SIMULA 67 (since 1986 simply SIMULA). In the present paper we will call it SIMULA 67 in order to distinguish it from the old simulation language that is not object-oriented.

It is interesting to observe that it was one of the authors of the OOP who emphasized the description role of the programming languages and namely of the OOP ones not too late after presenting the OOP paradigm to the world professional community. That was at a conference of Scandinavian mathematicians, namely by a paper (Dahl 1970), which was explicitly titled “Programming Languages as Tools for the Formulations of Concepts“. Note that Dahl tried to illustrate the descriptive function of “his own language” SIMULA 67 in the manner understandable for traditional mathematicians. He used such examples as notions common in plane geometry. The consequence was that in the illustrations the process orientation of SIMULA 67 did not come to light. Although it completely corresponds to conventional mathematics concepts that are rather simple, it also illustrates a deep gap between them and many concepts relating to complex dynamic systems. Nevertheless, SIMULA 67 was not only object-oriented but it preserved the process orientation, which had its predecessor, the simulation language SIMULA. Note that the synthesis of the object-orientation with the process orientation makes the language also agent-oriented.

If a set of classes is oriented to a certain branch, science or world viewing, the classes are connected. For example in geometry the class of circles has use the class of points, as the center of any circle is a point and moves of a circle can be defined by means of moves of its center and that it is meaningful and fruitful to introduce the relation that a point is inside a circle. Such a set can be viewed as a formal theory. SIMULA 67 allows aggregating such classes into another class called *main class*. One can include life rules methods and attributes into such a main class, similarly as one can connect a coordinate system with geometry. Thus classes can be exact representations of concepts as well as of theories handling with concepts. Therefore, in SIMULA 67 it is possible exactly define not only what means a factory, a machine, a train, a cell, a patient etc., but also what is a “theory” of production systems, of railway transport, of living organisms, of medical care etc., and it is also possible to specialize such theories, e.g. that of production systems to that of production systems of a certain sort, or – possibly – to a certain individual production system, in which many processes – mutually incompatible – may be under way.

2. INFORMATION PROCESSING OBJECTS

2.1. Characteristics

Information processing object (further IPO) exists in the physical world that is often modeled and namely simulated as existing in local time viewed as Newtonian. Beside of this, IPO handles with information that can be more or less distant (and separated) from the physical world where the object exists. A thinking human is an example of an IPO and a computer as well. The tendency of the informatization of the human society is to replace any human by something like a computer, especially in case the human should control a technological, social, military, biological and similar system. IPOs can be connected into information systems (further IS). The influence of common Newtonian time flow in such systems plays important role, but the main objective of IS consists in transferring information that may be independent of the common Newtonian time or not. In general, IS should process some information independent of its existence in the physical world but sometimes its ability to process information can be well applied also to optimizing its real time operation and similar tasks.

Therefore IPO can be viewed as composed of two levels, one level – let it be called *physical* – is related to its co-existence in common space and namely time with other objects (including other IPOs), and the other level – let it be called *information* – is related to its information processing function. In general, both the levels are mutually independent but in some exceptional but important cases they can be in contact.

From a strict ontological viewing, the physical level of an IPO cannot be neglected, as any IPO has to exist in space and in time in case it should perform its main

function (information processing), while it can exist but enter situations in that its main function is not used. If one considers IS as systems of IPOs, the neglecting of the physical levels of them may be dummy and lead to false prognoses (excepting the IS is evidently overdesigned).

2.2. Worries about Simulation of Information Systems

In the next considerations, let us suppose that IS is considered as a system containing IPOs. Note that then the importance of the physical level enters into IS because of the transmitting information among the IPOs. While that information is obtained at the information level of the concerned IPO (being hidden in it, as carried in its internal electronic processes), real time becomes essential for the next development in case the information is sent to another IPO.

The main reasons for troubles with decisions concern modifying one of the levels, conserving the other. Sometimes there is a working system *S* of IPOs designed to get a task different from the existing hitherto and the question is whether *S* will be able to satisfy that. Often the new task is the old enriched so that more computing is expected but sometimes the new task is rather different from the old one, e.g. when new laws for information processing came from outside or when a reorganization of an enterprise, a fusion or splitting components of it comes. Then the physical level remains but the information one is changed. It would be suitable to preserve all components of the simulation model that are images of the physical level.

On the contrary, sometimes the function of an IS should not be changed but the applied computers and their system are changed, e.g. in case of renovating the hardware. Then the information level should remain without changes while the physical level must be newly conceived. For the corresponding simulation, it would be dummy to demand reprogramming all what belongs to the information level.

Note that in modern information systems the particular IPOs make different tasks at the information level, as they may correspond to e.g. components of an enterprise, which in reality differ in great amount.

Summarized, the worries relate to two sorts of exact theories *P* and *J* where *P* reflects the physical level and *J* the information one, supposing that *P* can be modified without modifying *J* or vice versa, and – moreover – that in place of *J* different theories J_n ($n=1, 2, \dots$) may exist. The problem is that the theories *J* or J_i cannot be extracted from the world described in theory *P*.

For the next consideration, one can replace *J* by a set of J_n . In such a case J_n will be equal to J_m for n different from m and both will be equal to the original *J*. That enables to view any IPO to operate with its own theory describing its information level – let such a theory be called IPO’s *internal theory*.

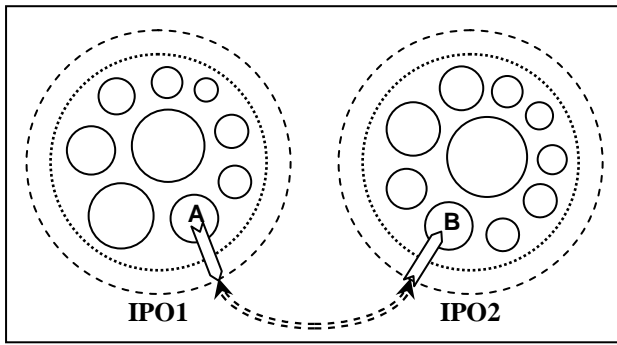


Figure 2: A Pair of IPOs

2.3. Solution of the Worries

The exact, natural and efficient solution of the mentioned worries should correspond to the description of the situation. It is illustrated in Fig. 2, where the number of IPO is reduced to 2: the world viewing at the physical level (two IPOs and a symbolic channel for physical signal exchange between them) is represented in dashed lines; the domain in that the operation of the information level is hidden is demarcated by a circle in dotted line, and the components of the description of the information level are represented as circles in full line. The full arrows represent the converters between the physical and information level data. In Fig. 2, an example of transferring a value from *IPO1* to *IPO2* is illustrated. On the left, an object of the internal theory, which is represented by circle *A*, gives its attribute for disposal to the IS; the value of the attribute is transferred from the electronic world of *IPO1* into the connection channel (or – in a human-being metaphor – from the mind of human *IPO1* to its mouth or other tool for communication), then the value is sent through the channel to *IPO2*, which converts it into the world described by its internal theory, namely as an attribute of object *B*. The conversion is expressed by the full arrow on the right and may represent a conversion from an input device into the internal electronics or – in case *IPO2* is a human – from his ears to his mind.

This mental scheme can be in principle described in a formal language that is not only object-oriented and process-oriented by also block-oriented. Block-orientation was introduced already in the language ALGOL 60 (Backus et al. 1960) that was not object-oriented, then well introduced into discrete event simulation language SIMULA (Dahl and Nygaard 1965 and 1966) that was process-oriented but not object-oriented and then successfully introduced into SIMULA 67 (Dahl and Nygaard 1968) (Dahl, Myhrhaug, and Nygaard 1968), which is commonly viewed as the first object-object oriented language but in fact is also process-oriented and block-oriented.

All entities (like classes and objects) of the physical level are to be aggregated into a block, that we can call **external block**. The life rules of the classes of the IPOs should contain a block called **internal block**, inside

which all entities of the information level should be introduced. Note that the other life rules of such classes may reflect phases in that the IPOs do not properly operate, namely during that they wait for signal. Because of the block hierarchy, any internal block has access to the attributes and methods of its porter (an IPO), which enables simple modeling of the conversion between the values of existing at the information level and the I/O channels. Fig. 2 can be considered as a scheme of the external block, the circles and the arrow in dashed lines belong to that block, the circles in dot line represent internal blocks and the circles in full line are entities belonging to the internal block. The access rules concerning the block orientation can be simply observed at the graphical scheme according to the following rule: an access can cross a boundary of any entity when going from inside, but cannot cross a boundary of an image of a block when penetrating from outside. A similar limitation is valid if an entity of any object is declared to be “protected” against such an access from outside.

The described paradigm admits a simple changing of blocks. In case the physically existing systems of IPOs should perform different information processing function than hitherto, one has only to replace the original internal blocks by new ones. In case the existing information processing should be preserved but installed at new hardware, one has only change the contents of the external block, which enclose the internal blocks.

Moreover, one can simulate systems containing IPOs that may perform different information processing during their existence, e.g. during different phases of day. Instead placing one internal block into a class of IPOs, one places more than one internal blocks there and – depending e.g. on the simulated time at the physical level – one programs the life rules of the class to switch among the internal blocks. It is also possible to enrich the simulation model of images of objects of the environment of the simulated IS (by introducing new objects at the physical level). One could also simulate interaction of several ISSs, when putting their descriptions at the physical level into one external block.

Another advantage of the described technique is as follows. One can expect encountering ISSs containing IPOs with similar (but not equal) information processing. The object-oriented languages that are also process-oriented and block-oriented enable to declare the common concepts of the concerned information processing activities as a main class and then to specialize it, so only the differences of the individual activities are explicitly expressed. The specialization can be iterated to reflect hierarchical ordering of the differences – see Fig. 3, where the dashed arrows represent specializations of the classes of IPOs and the dotted arrows represent specializations of the internal blocks: the entities (classes, objects etc.) belonging to the information level are represented as circles in full lines. The specialization starts from a class identified as *ABS*, because it may surely be an “abstract class”, no instances

of which are generated; it only figures as a common root of new classes so that

1. at one way class $C1$ is introduced as subclass of ABS (by adding entities represented as circles in doubled line),
2. at another way $C2$ is introduced as a subclass of ABS (by adding entities represented as circles in doubled line, which may be different from that added to $C1$), and
3. $C2$ itself is used for specialization, leading to class $C3$ that is subclass booth of ABS and $C2$ and at its information level using all tools disposed of $C2$ and – in addition – those outlined as circles in triple line.

3. CONCLUSION

The programming languages that are object-oriented, block-oriented and process-oriented can serve as excellent tools for description of the systems able to perform information processing concerning affairs rather distant from their own physics, structure and dynamics. Such languages represent rather new mathematical tools able to define new notions, including notions of information processing elements and their systems, but at the same time those languages figure as suitable programming languages, enabling a direct automatic modeling as the next step of the descriptions which uses them.

Note that those “triple-oriented languages” must satisfy a condition that was expressed in the last paragraph of subsection Mathematics and Simulation Languages: similarly as the simulation languages, the triple-oriented languages can figure as mathematical languages only in case they do not admit expressing

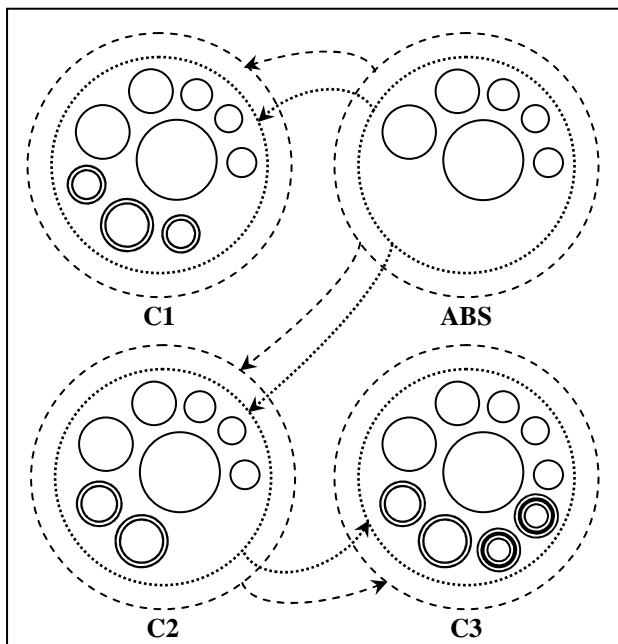


Figure 3: Hierarchy in Specializing

anything concerning the modeling computer: otherwise they are only programming languages.

At the University of Ostrava, the first experiments with simulation models of information systems were made (Kindler, Klimeš, and Křivý, 2007). The authors could exploit their experiences got during simulation of anticipatory production and logistic systems, like those described by Berruet, Kindler, and Coudert (2004), Kindler, Berruet, and Coudert (2003), Kindler (2000), Kindler, Coudert, and Berruet (2004), Kindler, Krivy, and Tanguy (2002), Krivy, Kindler, and Tanguy (2002), and Kindler, Krivy, and Tanguy (2003). SIMULA 67, mentioned above, appeared as a very suitable language, as described e.g. by Kindler (2004). Namely when a possibility appeared to identify individually every block, i.e. every model at any level (Krivy and Kindler 2007), SIMULA 67 was extremely appreciated by us. Beside this language, only BETA has the adequate properties (i.e. is triple-oriented and isolated from model hardware) but its syntax rules are very distant both from standard mathematical symbolism and from English vocabulary. Java is too joined with the hardware used for the model. The same holds for C++; another disadvantage of this language is absence of block-orientation.

ACKNOWLEDGMENTS

This work was supported by the Research Scheme MSM 6198898701 of the Czech Ministry of Education, Youth and Sport.

REFERENCES

- Backus, J. W. et al., 1960. Report on the Algorithmic Language ALGOL 60. *Numerische Mathematik*, 2: 106-136
- Berruet, P., Coudert, T., and Kindler, E., 2004. Conveyors With Rollers as Anticipatory Systems: Their Simulation Models. In: *Computing Anticipatory Systems CASYS 2003 – Sixth International Conference*, 582-592. August 11-16, Liege (Belgium).
- Buxton, J. N. (editor), 1968. *Simulation Programming Languages, Proceedings of IFIP Working Conference, Oslo, May 1967*, Amsterdam:North-Holland.
- Dahl, O.-J., 1970. Programming Languages as Tools for the Formulations of Concepts. *Proceedings of the 15th Scandinavian Congress*, 18-29, Oslo (Norway) 1968.
- Dahl, O.-J., Myhrhaug, B., and Nygaard, K., 1968. *Common Base Language*. Norwegian Computing Center: Oslo (Norway).
- Dahl, O.-J. and Nygaard, K. 1965. *SIMULA – A Language for Programming and Description of Discrete Event Systems: Introduction and Users' Manual*. Norwegian Computing Center:Oslo (Norway).
- Dahl, O.-J., and Nygaard, K., 1966. SIMULA – an ALGOL-Based Simulation Language. *Communications of ACM* 9:671-682

- Dahl, O.-J., Nygaard, K., 1968. Class and subclass declarations. *Simulation Programming Languages, Proceedings of IFIP Working Conference*, 158-174, May 22-28 1967, Oslo (Norway).
- Gordon, G., 1960. *Internal manual for simulator*, IBM, White Plains (New York):IBM
- Kindler, E., 2000. Nesting Simulation of a Container Terminal Operating With its own Simulation Model. *JORBEL (Belgian Journal of Operations Research, Statistics and Computer Sciences)* 40:169-181
- Kindler, E., 2004. SIMULA and Super-Object-Oriented Programming. In: Owe, O., Krogdal, S., Lychne, T., eds. *From Object-Orientation to Formal Methods*. Berlin:Springer, 165-182.
- Kindler, E., Berruet, P., Coudert, T., 2003. Conveyors with Rollers and Their Reflective Simulation. *International Workshop of Modelling & Applied Simulation MAS 2003*, 147-152. October 2-4, Bergeggi (Italy).
- Kindler, E., Coudert, T., Berruet, P., 2004. Component-Based Simulation for a Reconfiguration Study of Transitive Systems. *SIMULATION* 80:153-163
- Kindler, E., Klimeš, C., Křivý, I., 2007. Simulation Study With Deep Block Structuring. *Modelling and Simulation of Systems MOSIS 07*, 26-33. April 24-26, Ostrava (Czech Republic).
- Kindler, E., Krivy, I., Tanguy, A., 2002. Reflective simulation of logistic and production systems. *International Workshop on Harbour, Maritime and Multimodal Logistics Modelling & Simulation HMS, Modelling & Applied Simulation MAS 2002*, 97-102. October 3-5, Bergeggi (Italy).
- Kindler, E., Krivy, I., Tanguy, A., 2003. Simulation of Simulating Computerized Systems. *International Workshop of Modelling & Applied Simulation MAS 2003*, 16-21. October 2-4, Bergeggi (Italy).
- Korn, G., A., 1973. Recent Computer System Developments. *Proc. 7th Int. Conf. AICA*, 3-12.
- Krivy, I., Kindler, E., 2007. New Flexible Simulation Tool in SIMULA. *ESM'2007 – The 2007 European Simulation and Modelling Conference – Modelling and Simulation '2007*, 124-128. October 22-24, Ghent (Belgium).
- Krivy, I., Kindler, E., Tanguy, A., 2002. Software for Simulation of Anticipatory Production Systems. *IJCAS – International Journal of Computing Anticipatory Systems* 11:320-335
- Mitchell, E. E. L., Gauthier, J. S., 1976. Advanced Continuous Simulation Language. *SIMULATION* 26:72-78
- Strauss, J. C., et al., 1967. The SCi Continuous System Simulation Language. *SIMULATION* 9:291-303

AUTHORS BIOGRAPHY

Eugene Kindler studied mathematics at Charles University in Prague where he got a PhDr (doctor of philosophy) degree in logic and a RNDr (doctor of sciences) degree in mathematics. He worked as research specialist with Research Institute of Mathematical Machines in Prague (1958-66) and then as scientific specialist with Biophysical Institute of the Medical Faculty of Charles University (1966-73) and with Faculty of Mathematics and Physics (1974-2000) of the same University. Nowadays he is retired and collaborates as Professor with University of Ostrava in Czech Republic. He was visiting professor with the university at Pisa (Italy), at Morgantown (West Virginia), at Lorient and at Clermont-Ferrand (France) and foreign lecturer of Humboldt University (Berlin). Czechoslovak Academy of Science gave him CSc (Candidate of sciences) degree in physics and mathematics. E. Kindler was the author of the first Czechoslovak Algol compiler and of the first Czechoslovak digital simulation system. Since 1967, his main interests have been oriented to simulation (namely to simulation of simulating systems), the object-oriented programming and to applications in industry, life sciences and transportation.

IVAN KRIVY graduated at the Czech Technical University (1962) and the Charles University (1974). He worked subsequently at the Nuclear Research Institute (1962-1975), the Computer Art Establishment (1975-1977) and the Pedagogical Faculty of Ostrava (1977-1991). In 1991 he entered the Faculty of Science of the University of Ostrava and nowadays he works at the Computer Science Department as Professor in applied mathematics. Since 1992 his research activities are oriented to computer simulation and stochastic algorithms and their use in the global optimization. Krivy is a member of the American Mathematical Society, the Association of SIMULA Users, and the International Association for Statistical Computing. He is author of about 190 publications including 4 monographs.

CYRIL KLIMES graduated at the University of Technology (1976, Brno). In 1991 he was appointed by Assoc. Professor in informatics. He worked subsequently at the University of Mining of Ostrava, Czech Republic, the Educational Institute of the Czech Ministry of Fuel and Power Engineering and OKD-Control Automation Establishment. Nowadays he works as the head of the Department of Computers and Informatics and vice-rector for development and informatisation of the University of Ostrava. In 2005 he was appointed by Extraordinary Professor at Constantine the Philosopher University of Nitra (Slovakia). His research activities are oriented to information system engineering and process modeling.

SOLVING HIDDEN NON-MARKOVIAN MODELS: HOW TO COMPUTE CONDITIONAL STATE CHANGE PROBABILITIES

Claudia Krull^(a), Graham Horton^(b)

^{(a)(b)}University of Magdeburg, Department of Simulation and Graphics

^(a)claudia@sim-md.de, ^(b)graham@sim-md.de

ABSTRACT

Current production systems produce a wealth of information in the form of protocols, which is hardly exploited, often due to a lack of appropriate analysis methods. Hidden non-Markovian Models attempt to relieve this problem. They enable the analysis of not observable systems only based on recorded output. The paper implements and tests solution algorithms based on the original Hidden Markov Model algorithms. The problem of computing the conditional state change probabilities at specific points in time is solved. The implemented algorithms compute the probability of a given output sequence and the most probable generating path for Markov regenerative models. This enables the analysis of protocols of not observable systems and the solution of problems that cannot be solved using existing methods. This opens up new application areas, including a different approach to mining the wealth of data already collected today with huge monetary effort.

Keywords: hidden non-Markovian models, Hidden Markov Models, discrete stochastic models.

1. INTRODUCTION

Today machines and electronic components are getting more and more complex. As a result of that growing complexity they can produce a wealth of information on the system behavior in the form of protocols. These protocols are most of the time discarded or simply archived and not analyzed properly, often due to a lack of appropriate tools. Another problem is, that often the protocol entries are ambiguous. An error for example could have been produced by several internal causes and each cause could also result in different errors. If that is the case, finding the exact or even a probable system behavior that produced the given protocol is almost impossible using existing methods.

The recently developed paradigm of Hidden non-Markovian Models (HnMM) strives to relieve this problem. HnMM are an extension of Hidden Markov Models to more general discrete stochastic hidden models. HMM can analyze hidden behavior based only on recorded system output. However, the hidden model is a discrete-time Markov chain and therefore severely limited. The extension to discrete stochastic hidden models enables more realistic models and the analysis

of real life systems. These HnMM can in theory analyze the recorded protocols of not observed systems and determine the protocol probability or the most likely generating behavior.

Solving HnMM is still an issue of current research. HnMM can be analyzed using Proxel-based simulation (Lazarova-Molnar 2005, Wickborn et al. 2006, Krull and Horton 2008), which is however quite expensive, because it computes the complete reachable model state space and consequently all possible generating paths. An alternative solution method was proposed in (Krull and Horton 2009), where the theory for HnMM was described as well as a theoretical transfer of the original HMM solution algorithms to the new paradigm.

This paper shows an implementation of the adapted HMM algorithms. It also solves the problem of how to compute the conditional state change probabilities at a given point in time in continuous time. This is not trivial, since in theory the probability of a state change via a continuous distribution function at a specific point in time is 0. Furthermore the performance of the implemented algorithms is tested under growing model size and growing protocol length. A comparison with the current Proxel-based solution algorithm will show the differences of the two approaches. The adapted HMM algorithms are much faster and enable the analysis of longer sequences and larger models. However, the algorithms only compute the cumulative state probability or the one most likely generating path. Therefore both solution methods have different application areas and should be developed further to be applicable to real world problems.

1.1. Possible Application Areas of HnMM

Application areas encompass all engineering problems where the true system or machine state is not detectable or has not been stored and only protocols containing ambiguous records are available. A possible application area outside of engineering could be the analysis of the symptoms of a patient, which can be interpreted as the output of the unobservable true state of his or her disease.

Today in some parts of the industry a huge effort is invested in collecting data on existing production lines by installing internal sensors and logging the observed processes. The resulting amount of data is often huge

and too often not consistent, since the calibration and coordination of different sensors can be very difficult. Consequently the analysis of the data can be tedious. Often the effort that would need to be invested to gain reliable sensor equipment and data analysis is larger than the expected benefit of such a step.

A concrete industry application of HnMM could therefore be the following. A small production line is being observed over a limited period of time through external sensors such as cameras or photoelectric barriers. These sensors can only detect the visible part of the production process and might not be able to observe the individual machines' or products' status or a hidden buffer content. The analysis of these logs can help determine the working status of the machines and their current performance indices, helping the owner of the production line in his decisions regarding the modernization or replacement of machines. Hidden models are necessary in this context, since only a portion of the production process can be observed. However, HMM are not sufficient, since a production process is most likely time dependent and includes several non-Markovian transitions. Only the extension to Hidden non-Markovian Models can provide a time dependent model for the hidden process.

The application of the proposed method and HnMM analysis in this example could shift and considerably reduce the necessary investment for sensor equipment and resulting effort for data collection, enabling also smaller firms to conduct such studies.

2. STATE OF THE ART

2.1. Hidden Markov Models

Hidden Markov Models (HMM) (Rabiner 1989, Fink 2008) are a well known paradigm in the realm of speech and pattern recognition. They are used to determine the hidden meaning of an observed output in the form of recorded speech or scanned pictures or letters. HMM consist of a discrete-time Markov chain (DTMC) (Bolch et. al 2006) as hidden model and a second stochastic process emitting symbols with given probabilities based on the current system state. Given a sequence of output symbols, one can solve the following three tasks:

- Evaluation: Determine the probability of that sequence for a given model.
- Decoding: Determine the most likely generating state sequence for a given model.
- Training: Train an initialized model to produce that sequence most likely.

Algorithms exist to solve all of these problems. The evaluation and decoding problem are solved through iterative algorithms. Training is by far the most complex task of the three and is usually solved through an optimization-like method.

The applicability of HMM to our area of interest is limited due to DTMCs as hidden model. These can only

accurately represent memoryless processes and are therefore not very realistic for most applications. Current extensions of HMM focus on increasing the speech recognition capability (Fink 2008). Most of them change the output process from discrete to continuous. The few attempts to generalize the hidden process of an HMM were abandoned due to the minimal increase in speech recognition capability, compared to the large increase in computational complexity of the solution algorithms.

2.2. Hidden non-Markovian Models

The extension of HMM to more general models leads to so-called Hidden non-Markovian Models (HnMM). These have been formalized in (Krull and Horton 2009) and the difference to HMM is twofold. HnMM use the state space of a discrete stochastic model (DSM) as hidden model, enabling non-Markovian state transitions. Furthermore, they associate the symbol emissions with the state changes, since these are most often the objects of interest in a DSM.

The Proxel-based simulation algorithm was proposed as HnMM solution method even before formalization (Wickborn et al. 2006, Krull and Horton 2008). The Proxel algorithm is a state space-based simulation method that discovers all possible system developments in discrete time steps (Horton 2002, Lazarova-Molnar 2005). These possible development paths can correspond to generating paths of HnMM output sequences. The solution of HnMM via Proxels is however quite expensive. The method explores all possible generating paths and has therefore a large memory and runtime requirement and is limited to very small models.

The goal of this paper is to implement the adapted HMM algorithms presented by Krull and Horton (2009) that promise to be more efficient. One major challenge is the computation of conditional state change probabilities at given points in time in continuous time. This would enable the analysis of larger and therefore more realistic models in order to increase the practical applicability of HnMM.

3. ADAPTED HMM ALGORITHMS

3.1. Existing HnMM Theory

This section gives a brief definition of HnMM and their components. A more detailed formalization can be found in Krull and Horton (2009). A Hidden non-Markovian Model contains the state space of a discrete stochastic system as hidden model. The symbol outputs are associated to the transitions.

$$(S, C, V, A, B, \pi) \quad (1)$$

An HnMM is represented by a 6-tuple as seen in Equation (1). The first three elements are the set of discrete system states S , the set of state changes C , the set of output symbols V . The transition behavior is described by matrix A , which contains the instantaneous

rate functions (IRF) (Horton 2002) of the state transitions. The symbol output process is described by B , where each state transition is associated with output symbols, each with its own emission probability. The initial probability vector π contains the probability to be in each of the discrete system states at the start of the analysis.

For HnMM analysis one also needs a representation for the observed symbol sequence O and a sequence of state changes Q . Both sequences contain elements associated with a time stamp (see Equations (2) and (3)).

$$O = \{(o_1, t_1), (o_2, t_2), \dots (o_T, t_T)\} \quad (2)$$

$$Q = \{(c_1, t_1), (c_2, t_2), \dots (c_T, t_T)\} \quad (3)$$

In the paper Krull and Horton (2009) also formalized the adaption of the original HMM solution algorithms to HnMM. However, this was only possible for Markov regenerative processes that regenerate at every firing of a transition. Nevertheless we hope that for that class of models the algorithms will show much better performance than Proxel-based HnMM analysis. In the current paper we also concentrate on models where all transitions emit symbols, and where therefore each state change is represented by a symbol emission. This model configuration poses the easiest one and should therefore be implemented first.

3.2. Computing the Conditional State Change Probability

The key challenge when implementing the adapted HMM algorithms is the computation of the conditional state change probability. Equation (4) describes this probability of making the state change from state s_i to state s_j at time t_n under the condition of emitting symbol o_n . The computation of this quantity is not straight forward, since the point in time of the state change is predetermined by the time stamp in the symbol sequence. In theory, the probability of a state change at a specific point in time via a continuous distribution function is 0. This can however not be the solution for the given case, because we know that the state change will happen at exactly that point in time.

$$P(c_{ij} \text{ at } t_n | o_n) = p \quad (4)$$

The solution we propose is to compute the probability to remain in state s_i until time t_n and then assume a total state change probability of 1, meaning that all of the remaining probability is that of the state change. The current rate of a state change that has not occurred yet can be described by the instantaneous rate function (IRF) (Horton 2002). This can also be used to compute the probability to not perform a state change within a certain time period. The exact remaining probability can be computed using a parallel ODE45 (Buchholz 2008) integration scheme for the transitions IRF (or the in Markov regenerative case the transitions cumulative distribution function (CDF)). If at the time

of the symbol emission there are multiple state changes possible, the remaining probability is weighted with the normalized IRF of each transition (see Equation (5)).

$$p = P(\text{stay in } s_i \text{ until } t_n) * \frac{\mu_{ij}(t_n - t_{n-1})}{\sum_k \mu_{ik}(t_n - t_{n-1})} \quad (5)$$

This measure should be a good estimate of the conditional state change probability at a specific point in time. The experiments will show its practical applicability.

3.3. Implementation of the Adapted HMM Algorithms

The HnMM solution algorithms based on the original HMM algorithms were described in Krull and Horton (2009). The implementation of these algorithms was straight forward, once the problem of computing the conditional state change probability had been solved.

The Forward algorithm and the Backward algorithm (Rabiner 1989) solve the evaluation problem for HMM. Their general structure could be transferred directly to HnMM (see Krull and Horton (2009) for further details). The conditional α and β measures were computed for every time stamp and the final trace probability derived.

The Viterbi algorithm solves the decoding problem for HMM. It is a greedy algorithm that computes the most likely generating path for a symbol sequence. The conditional γ probabilities were also computed for every time stamp. The backtracking yielded the most likely generating sequence in terms of state changes that happened. This is actually the most likely sequence, since we are dealing with a Markov regenerative model that regenerates at every firing of a transitions. When lifting that restriction, the optimality of the path discovered can no longer be guaranteed.

The training algorithm was not attempted in this implementation. Already in the formalization paper it was stated, that it is unclear how to train non-Markovian state transitions. Therefore this HMM problem needs to be tackled otherwise. One method was already described in Isensee et al. (2006). There, discrete phase-type distributions were used to turn a non-Markovian model into a DTMC, which was then trained using the original Baum-Welch algorithm. We will therefore not investigate the problem of model training further in the current paper.

The current implementation now enables the analysis of some classes of HnMM through the adapted Forward and Viterbi algorithms.

4. EXPERIMENTS

This section describes some experiments conducted to show the correct functionality of the adapted HMM algorithms, to compare it with the Proxel-based solution algorithms and to test its performance.

The model topology used for the experiments is a machine maintenance model with three states. The hidden model part in the form of a stochastic Petri net is

shown in Figure 1. The machine can be in working condition (*OK*), under maintenance (*Maint*) or failed (*Failed*). The transitions between these states are non-Markovian except for the failure distribution and every transition can emit symbols.

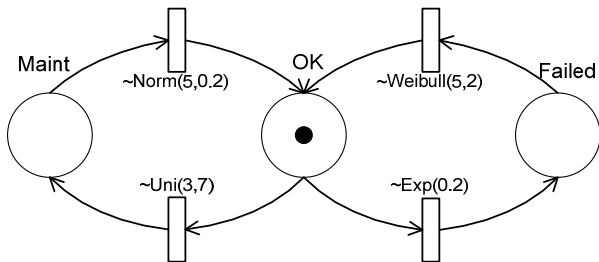


Figure 1 : Example 3-State Model Hidden Part

The complete HnMM consists of the following elements ($S, C, V, A(t), B, \pi$). A full description of the notation can be found in Krull and Horton (2009).

$$S = \{OK, Maint, Failed\}$$

$$C = \{MaintenanceInterval, Maintenance, Failure, Repair\} = \{MI, M, F, R\}$$

$$V = \{A, B, C, D, E\}$$

$$A(t) = \begin{bmatrix} 0 & \mu_{MI}(t) & \mu_F(t) \\ \mu_M(t) & 0 & 0 \\ \mu_R(t) & 0 & 0 \end{bmatrix}$$

$$B = \begin{bmatrix} 0.1 & 0.2 & 0.2 & 0.4 & 0.1 \\ 0.3 & 0.1 & 0.2 & 0.1 & 0.3 \\ 0.1 & 0.3 & 0.1 & 0.2 & 0.3 \\ 0.4 & 0.2 & 0.1 & 0.1 & 0.2 \end{bmatrix}$$

$$\pi = (1,0,0)$$

The output sequences consisted of random series of the five output symbol letters of length 10 to 100,000. The time stamps were equally spaced at intervals of 5 time units.

4.1. Validation Experiments

The first validation experiment compared the resulting trace probability and Viterbi path of the adapted HnMM algorithms with the Proxel-based HnMM analysis. Both algorithms can handle non-Markovian distributions and symbol outputs associated with the transitions. Therefore the same model was used for both algorithms. A symbol sequence of length 20 was used as benchmark.

Both the adapted HMM algorithm and the Proxel algorithm determined the same most likely path of state changes to have produced this sequence. This is the most important fact, showing the equivalence of both approaches. The total probability of the trace was in the same order of magnitude for both algorithms. An exact match could not be obtained due to the Proxel algorithm using a discretization time step of 1 time unit, leading to an approximate result. Furthermore, the absolute probability values are not as important as the difference in values obtained for different traces or different generating paths, using the same tool. These help

discriminate among different traces and paths to choose the most likely one.

The second validation experiment compared the results of the adapted HMM algorithms to the results of the original HMM algorithms. For that experiment an HnMM had to be adapted to mimic the behavior of an HMM emitting a symbol in every step. This included using only transitions with exponential distributions. Since the resulting HMM and HnMM model only correspond approximately to each other, the best match that could be achieved here was also the same most likely generating state change sequence (HnMM) corresponding to the Viterbi path (HMM).

These two experiments showed that the adaption of the HMM algorithms did not affect the main functionality in a serious way, so that they still compute valid results at least in terms of the most likely generating path.

4.2. Performance Experiments

The second set of experiments was conducted to obtain information on the parameters influencing the performance of the adapted HMM algorithms. The machine maintenance model and symbol traces of a maximum length of 100,000 were used. Furthermore was the model size varied by increasing the number of states. This was done by replicating the machine maintenance model and concatenating the states. The measure of interest was the combined *Forward* and *Viterbi* algorithm runtime in seconds on a laptop with an Intel® Core™2 Duo CPU with 2 GHz and 2GB RAM running Windows XP.

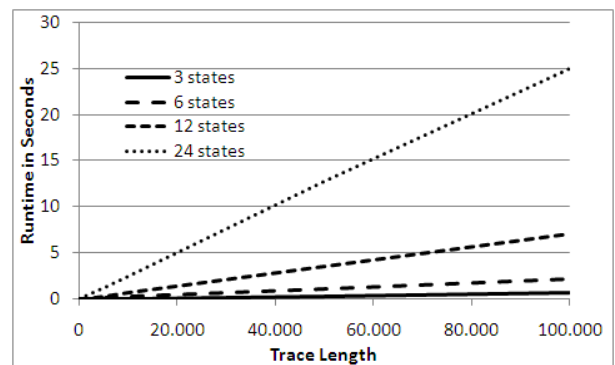


Figure 2 : Runtime over Trace Length for Different Model Sizes

Figure 2 shows the development of the algorithm runtime with increasing trace length for different model sizes. All four graphs show a linear increase in runtime with increasing trace length. This behavior is as expected. It corresponds to the linear relationship of runtime to trace length for the original HMM algorithms. The memory requirement was not tested and compared here, since due to the iterative nature of the algorithms the amount of data that needs to be stored in working memory is constant for every step and only marginally increases with growing model size.

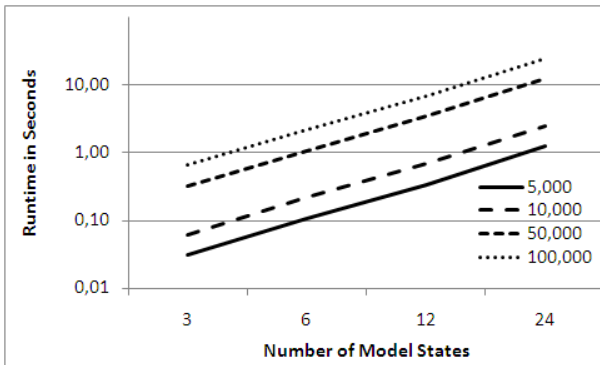


Figure 3 : Algorithm Runtime (logarithmic scaling) over Model Size for Different Trace Lengths

Figure 3 shows the same data from a different perspective. Here the runtime of the algorithms is shown in relation to the size of the model for four traces of different length. The runtime is depicted in logarithmic scaling. One can see that with growing model size, the runtime grows faster than linearly but somewhat exponentially. This is due to the increased complexity of the model that increases the amount of computation time needed at every step.

Another experiment was conducted testing the influence of the number of output symbols on the algorithm performance. Since increasing the number of output symbols did not affect algorithm performance, the results are not included here.

Overall the runtime behavior of the adapted HMM algorithms is as expected very similar to that of the original HMM algorithms. However, the absolute speed of the computation is very fast, even with growing model sizes. This promises a good practical applicability.

4.3. Comparison Experiments

The last set of experiments compared the results and performance of the adapted HMM algorithm with that of the Proxel HnMM analysis. As seen above, the runtime cost of the adapted HMM increases linearly with the sequence length. The runtime cost and memory requirement of the Proxel method on the other hand increases exponentially with increasing sequence length, mostly due to a growing number of possible generating paths that need to be handled in every step.

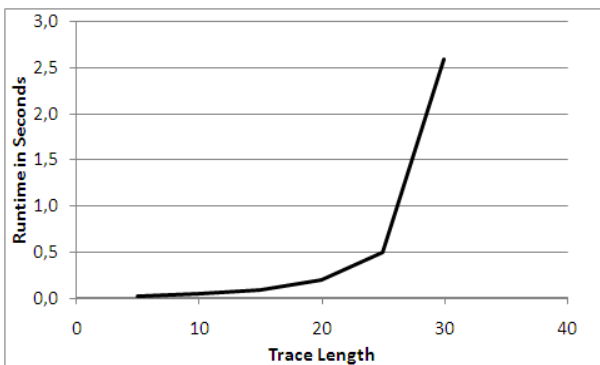


Figure 4 : Proxel HnMM Analysis Algorithm Runtime over Trace Length

The performance of the Proxel method is directly influenced by the discretization time step, which has no equivalent in the adapted HMM algorithms. Therefore the Proxel algorithm is inherently slower and more memory consuming than the adapted HMM algorithms. This results for example in a runtime of several seconds for the above three state model with a sequence length of 30 (see Figure 4), which is already several orders of magnitude slower than adapted HMM. Using significantly longer traces, the Proxel algorithm aborted without a result, because of too many recursions. Adapting the Proxel algorithm such that it can handle longer traces is one area of future research.

The results of the Proxel method on the other hand are more rich than those of the adapted HMM algorithms. The Proxel method computes in one run the total trace probability as well as all possible generating paths above a certain cutoff probability. The adapted Viterbi algorithm only returns the one most likely generating path. A ranking of all or at least of the top X paths enables their comparison to gain valuable insights. A situation where the top paths have almost equal probability, making them equally likely, would not be apparent using the adapted Viterbi algorithm.

The last point is the applicability of the methods. The adapted HMM algorithms are right now only applicable to Markov regenerative models that regenerate at every firing of a transition, and where all transitions emit symbols. This limits the model class that can be analyzed using the proposed algorithms significantly. The Proxel method on the other hand does not have these restrictions, it can handle transitions that do not emit symbols and time dependent transitions whose age is not affected by the firing of other transitions. This makes the Proxel method in general the method of choice for a given model. However, if the model is of the type that can be analyzed using adapted HMM, then these algorithms should be preferred, due to the superior runtime behavior.

5. CONCLUSION AND OUTLOOK

The paper documents the implementation of solution algorithms for Hidden non-Markovian Models. The algorithms implemented are an adaptation of the original Hidden Markov Model solution algorithms. One crucial issue was the computation of the conditional state change probability in continuous time, which was solved in this paper. The implemented algorithms are however limited to Markov regenerative models that regenerate at every state change.

The algorithms are much faster than the Proxel solution proposed in earlier papers. However, the results are far less general. It is now possible to find the probability and generating path for a given system protocol. Using the current algorithm, significantly larger models and longer traces can be analyzed then using Proxels, and therefore more realistic problems are possible.

Future work includes the extension of the current implementation to further model classes. That will involve models where not all possible state changes generate output symbols, which will probably result in a mix of the adapted HMM algorithms and the Proxel solution method for computing the path probability between two consecutive symbol emissions.

Competitive algorithms for the analysis of HnMM will enable the solution of questions which cannot be solved using existing tools today. Protocols can be used to determine possible system behaviors that could likely have produced them. This can in turn help to determine whether a given system is functioning within the given specifications or, in the medical case, in what state of a disease a patient is in most likely. Eventually, HnMM can contribute to mining the wealth of data which is already being collected in current production systems and elsewhere, decreasing the necessary effort for obtaining valuable results.

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.
- Buchholz, R., 2008. *Improving the Efficiency of the Proxel Method by using Variable Time Steps*. Master's thesis, Otto-von-Guericke-University Magdeburg.
- Fink, G.A., 2008. *Markov Models for Pattern recognition*. Springer-Verlag Berlin Heidelberg.
- Horton, G., 2002. A new paradigm for the numerical simulation of stochastic petri nets with general firing times. *Proceedings of the European Simulation Symposium 2002*. SCS European Publishing House.
- Isensee, C., Wickborn, F., Horton, G., 2006. Training Hidden Non-Markov Models. *Proceedings of 13th International Conference on ANALYTICAL and STOCHASTIC MODELLING TECHNIQUES and APPLICATIONS*, Bonn, Germany, pp. 105-110.
- Krull, C., Horton, G., 2008. The Effect of Rare Events on the Evaluation and Decoding of Hidden non-Markovian Models. *Proceedings of the 7th International Workshop on Rare Event Simulation*, pp. 153-164.
- Krull, C., Horton, G., 2009. HIDDEN NON-MARKOVIAN MODELS: FORMALIZATION AND SOLUTION APPROACHES, *Proceedings of 6th Vienna Conference on Mathematical Modelling*, Vienna, Austria, pp. 682-693.
- Lazarova-Molnar, S. 2005. *The Proxel-Based Method: Formalisation, Analysis and Applications*. PhD thesis, Otto-von-Guericke-University Magdeburg, Germany.
- Rabiner, L. R., 1989. A tutorial on hidden Markov models and selected applications in speech recognition. *Proceedings of the IEEE*, vol. 77, pp. 257-286.
- Wickborn, F., Isensee, C., Simon, T., Lazarova-Molnar, S., Horton, G., 2006. A New Approach for

Computing Conditional Probabilities of General Stochastic Processes. *Proceedings of 39th Annual Simulation Symposium 2006*, Huntsville, USA, pp. 152-159.

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.

USE OF SIMULATION IN EGOVERNMENT PROCESS DEVELOPMENT. A CASE STUDY USING THE SIMULATION TOOL SIGHOS.

Y. Callero^(a), R. Aguilar^(b)

^{(a)(b)}Department of Systems Engineering and Automation, and Computer Architecture.
University of La Laguna. Spain

^(a)ycallero@isaatc.ull.es, ^(b)raguilar@ull.es

ABSTRACT

Simulations are commonly used in business process reengineering. This paper demonstrates their applicability to systems where human factor is important like eGovernment processes. It will discuss an example of a Canary Islands government process where the effects of a reorganization is simulated using the simulation tool SIGHOS. Through the simulations it is possible to monitor the process and acquire data which is otherwise impossible to obtain.

Keywords: discrete event simulation, eGovernment, SIGHOS, eBusiness process reengineering, business process reengineering.

1. INTRODUCTION

Technological change has catalyzed organizational changes in several fields over the years. Focusing in the public administration, new technologies was appearing and replacing older formats and processes since the government began using telegraphs and telephones. Those actions caused different kinds of benefits like minimization on process execution time, reduction on resources cost, etc. Nowadays another technological change is arising. The growth of the architectures, services and applications around Internet is causing social and cultural changes over the world. In addition, another organizational change is catalyzed, the eGovernment.

Scholl (Scholl 2003) notice that there isn't a single definition of eGovernment widely accepted. Some definitions are centered in the area of impact, others on the reorientation on citizens', business' and agencies' needs or on gains in administrative efficiency. In 2003, the European Union acknowledges the fact of the new organizational change (European Commission 2003) and defined a roadmap to improve then eGovernment incorporation. In that roadmap, the eGovernment was defined by "*the use of information and communication technologies in public administrations combined with organizational change and new skills in order to improve public services and democratic processes and strengthen support to public policies.*"

The reasons to improve eGovernment policies were made activities more efficient and more effective. The efficiency will be reached by the bureaucracy's reduction and the effectiveness will be reached making the process accessible, user-friendly, secures and targeted. These improvements will boost economic growth throughout the economy as a whole (European Commission 2003).

Scholl (Scholl 2003) identifies three different areas for the eGovernment introduction: government to government (g2g), government to business (g2b) and government to citizen (g2c). Government to government is the online non-commercial interaction between government organizations, and authorities and other government organizations, departments, and authorities. Government to business is the online non-commercial interaction between local and central government business sector. Government to citizen is the online non-commercial interaction between local and central government and private individuals, rather than the commercial business sector.

Focusing in these three areas, the EU started several action plans. One of them was the "The eEurope 2005 Action Plan" (European Commission 2003). The principal goal was the widespread availability and use of broadband by 2005. When eEurope 2005 finished, the EU started another action plan: i2010 (European Commission 2006). It promotes the positive contribution that information and communication technologies (ICT) can make to the economy, society and personal quality of life. These action plans identified twenty basic electronic public services and four development phases for each service. They are used as indicators of the proximity to the final goal. In 2008, the average of the basic services implementation for EU-27 members was 59% and the average of development phases was 77% (European Commission 2003). In particular, Spain is over the UE average in both cases. Spain implements 70% of the basic services and the development phases is 84%. These data place Spain out of the ten members with the highest implementation levels so there are several areas to improve on the eGovernment implementation.

In this paper we present a case study of the use of simulation modeling in a g2c process development. We analyze the process of Canary Islands Registry of Associations. We compare the existing method of performing the process with the new organization scheme for the electronic process. Simulation is used to compare old a new process. To execute the simulation models a discrete event business process simulation tool, called SIGHOS (Aguilar et al. 2006), is used. Our research shows that the use of simulation is a valid way to predict the effects of the new electronic process incorporation.

In section 2 a summary about differences of business reengineering and eBusiness reengineering is presented. The use of simulation models in these disciplines are mentioned too.

In section 3 some examples of business process simulation and several cases where the simulation is used to implement eGovernment reorganizations are presented.

In section 4 a specific case study where the University of La Laguna's (ULL) simulation group uses simulation in the eGovernment implementation in Canary Islands.

2. EBUSINESS PROCESS MODELING AND SIMULATION MODELING

Business process re-engineering (BPR) was used to rationalize process and products since the eighties and nineties of the last century as seen in Janssen et al. (Janssen et al. 2003). The automation of the production activities and the competition, between companies, to resolve the customers' demands was the principal reasons of the growth of these disciplines. One of the techniques used in BPR was the business process modeling (BPM). Kovacic and Pecek (Kovacic et al. 2007) enumerate the aims of using BPM:

- To help the business process innovation team obtain a holistic view of the process being studied.
- To identify areas of improvement.
- To visualize the impacts and implications of new processes.
- To describe the rules that underlines the business process.

In fact, business process models are maps or images of the logical and temporal order of business activities performed on a process object. BPM has been adopted as an appropriate way to describe business behavior.

Nowadays the situation has drastically changed. Customers have become increasingly demanding and product innovation rates are high. Also, globalization of markets and the availability of new electronic media increased the new players in existing markets. This fact causes a high pressure to work more effectively, reduce costs and increase the need to be more flexible. The appearance of eBusiness and eGovernment has changed

the way that organization process operates. Janssen et al. (Janssen et al. 2003) describe that business process reengineering and eBusiness process reengineering (EBR) share many of the same characteristics but there are some inherent differences between them. These differences are focused in the behavior interpretation of each process more than in a description of new elements to take into account. Therefore, focusing in the modeling aspect, the differences detected in eBusiness processes don't generate new patterns above defined Aalst et al (Aalst et al. 2003). So an eBusiness can be modeled using a business process modeling tool based on this patterns.

Using models which describe the eBusiness behavior it's possible to simulate it to obtain information about the process. Simulation has an important role in BPR. It enables quantitative estimations to be made on the influence of the redesign process on system performances. Kovacic and Pecek (Kovacic et al. 2007) enumerate several reasons for introducing simulation modeling into process modeling:

- Simulation allows for the qualitative of process dynamics.
- The influence of random variables on process development can be investigated.
- Re-engineering effects can be anticipated in a quantitative way.
- Process visualization and animation are provided.
- Simulation models facilitate communication between clients and the analyst.

In this regard, several techniques and tools are improved. IDEF0, IDEF3, Petri Nets, System Dynamics, Knowledge-based Techniques and Discrete Event Simulation are only some examples of widely used business process modeling techniques. The list of available business process modeling tools supporting simulation is growing year by year. In this way, the ULL's simulation group is engaged in the development of a discrete event simulation tool based on EPC (Event-Driven Process Chain) notation called SIGHOS (Aguilar et al. 2006). This tool was used to model and simulate several business process obtaining results which was used to analyze the processes behavior and simulate possible changes.

3. CASES OF BUSINESS PROCESS SIMULATION. E-GOVERNMENT SIMULATION

There are areas which have process in which people play the main role like health care, the service industry, traffic modeling and many others. Usually, these processes are unpredictable. In these cases, simulations are becoming increasingly applied to obtain information about the behavior of the process. There are several examples of the use of simulation in BPR in the literature.

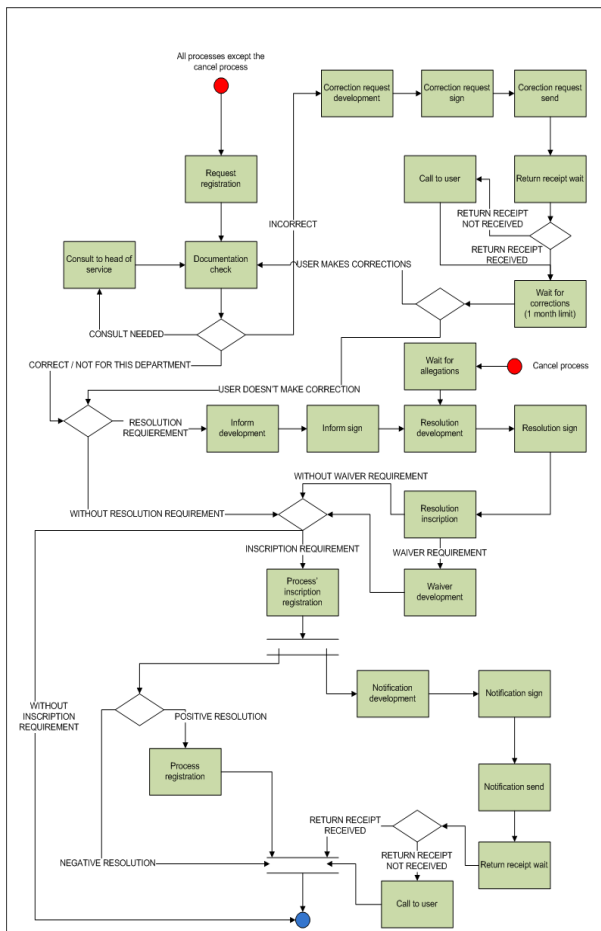


Figure 1. Registry of Associations workflow.

Albores et al. (Albores et al. 2002) use a discrete-event simulation to model an Internet based ordering system of a mineral water manufacturer. The scope of the project is to explore changes in the way the Purchase Order for Haulier (POH) is sent to the hauler. Kumar and Phorommated (Kumar et al. 2006) used simulations of a redesigned sheeting operation in a setup reduction process at a pulp and paper manufacturer. Nikolaidou et al. (Nikolaidou et al. 2001) presented a study of modeling and automating business processes of a medium sized bank introducing Internet technologies. In this case, the authors used a discrete event simulation tool. Moon and Phatak (Moon et al. 2005) showed that the limited planning functionality of the Enterprise Resource Planning (ERP) system can be complemented by discrete event simulation models. Greasley (Greasley 2005) analyzed a business process approach to a change in custody of prisoners' process. Ozbay and Bartin (Ozbay et al. 2003) studied an incident management situation with the Arena simulation.

Focusing in ULL's simulation researching group, Aguilar et al. (Aguilar China et al. 2009) studied a case in hospital management using discrete event simulation. Baquero et al. (Baquero et al. 2005) presents the simulation of an IT Center for user's attention.

The uses of simulation tools in eGovernment process reengineering are described in the scientific literature too. Kovacic and Pecek (Kovacic et al. 2007) presented the modeling and analysis of the procedure for obtaining social help in Slovenia. Li (Li 2008) described two real cases which were chosen to investigate how the e-government incorporation affected the organization in the Chinese government. The cases were about tax and urban hygiene. Janssen and Cresswell (Janssen et al. 2005) presented a case focused on modeling of an information system to support an integration operation called "business counter". This operation is used by the governments to provide a mix of services to business. Wu and Hu (Wu et al. 2007) proposed a multi-agent modeling and simulation approach using E-Government Group Behavior Model to research complex group behavior in E-government implementation.

As seen in this section, lot of cases has used simulation to obtain answers about specific problems. In this way, the ULL's simulation group used the business process modeling and simulation tool called SIGHOS to quantify the benefits of incorporate electronic processes in the Canary Islands government. In the next section a specific case study will be presented. This case is focused on the Registry of Associations process that will be redefined to incorporate electronic processes in order to improve it working system.

4. CASE STUDY

Several public administration processes are reorganized into electronic processes in Canary Islands government nowadays. The government's managers wanted to qualify the benefits of each reorganization so called to the ULL's simulation group to resolve that problem. A specific simulation case study is exposed, the Registry of Associations.

4.1. Registry of associations system

The Spain and autonomous regions governments share the competences about the associations' control. The autonomous region government takes all responsibility for the establishment of different ways to set up, manage or dissolve associations. The Registry of Associations was defined to manage all of these operations. Focusing in Canary Islands region, each province has its own Registry of Associations head office. So exists one head office in Las Palmas province and another in Santa Cruz of Tenerife province. Each of these head office has its own staff. In the one hand, Las Palmas province has two government employees and two technicians. In the other hand, Santa Cruz province has three government employees and one technician. In addition, exists one head of service and one general director for the entire autonomous region.

This staff takes care of the execution of all Registry of Associations' activities. The Registry of Associations has nineteen different activities which logical organization is showed in the workflow of

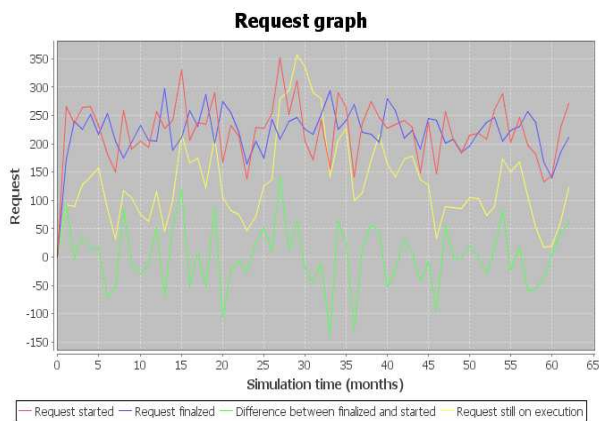


Figure 2. User request graph.

Figure 1. The activities execution is started by a customer process solicitation. The activities execution flows depends on the execution process characteristics. There are different decisions points in the workflow which route the flow depending on those characteristics.

Sixteen different processes can be executed in Registry of Associations' workflow. In addition, each of these processes can have different characteristics like if the documentation presented is correct or if the process not corresponds to the Registry of Associations and will be redirected to another department.

4.2. Registry of Associations simulation model generation

The business process model generation is based on the correct definition of several elements of the process:

- The resources which get involved in the process execution.
- The activities which defined the resources work and the relationship between them.
- The workflow which defines the order where the activities are executed.
- The elements which started the process execution.

4.2.1. Resources modeling

The resource's types are defined in the Registry of Association introduction. There are four types: government employees, technicians, head of service and general director.

After several meetings with Canary Islands government responsible and the company which is developing the electronic process, they indentified that the government employees and the technicians share they work. So they can be considered the same resource type which will be indentified with the name worker. Workers start their work at 8:00 AM and finalize it at 2:00 PM that represents seven hours of work every day. But, in the model, only working days are considered so the public holidays are needed to take into account. Erasing public holidays, workers work six hours per day all the year. For the head of service and general director the working hours are the same as workers.

After all, in the model, are defined for workers for each province and one head of service and general director for the entire autonomous region. All of them work six hours per day starting at 8:00 AM.

4.2.2. Activities modeling

In the Registry of Associations process are identified nineteen activities which resources executes. Thirteen of them are executed by workers, three by the head of service and one by the general director. The remaining activities represent inactivity time waiting for a customer action. These waiting activities are the documents corrections presentation or the arrival of one notification return receipt.

There isn't any data about the duration of the activities. The data that government can provide are:

- The corrections waiting time limit: 1 month. In addition the government quantifies the percentage of users that made corrections in 60%.
- The head of service and the general director only executes sign activities. The problem is that they can only sign documents for a province where they physically are in that province. Usually they are three days in Santa Cruz de Tenerife province and two days in Las Palmas province.

The assumptions to define the activities execution time are:

- There are some activities which execution time can be inferred by the practical experience. In case of documents corrections, the assumption was that the average of time to present the correct documentation is a half of the limit time. It is around fifteen days with a standard deviation of four days. In case of the waiting time for a return receipt we can assume the usually delay of the Spanish postal service in Canary Islands. In this case two days with a standard deviation of one day. In case of a phone call to customers, workers don't spend much time doing that. They focused the talk in the info that has to give to the customer. The execution time for this activity is considered around five minutes.
- The activities which represent documents signs by the head of service or by the general director had specific assumptions. The waiting time for a document sign was considered. Taking into account that Santa Cruz province documents are signed three days per week and Las Palmas province documents are signed two days per week the waiting average time for each document sign can be calculated. This time is used as execution time for the head of service and general director sign activities.
- The execution time of the remaining activities is calculated using the average number of processes which are resolved in one month. It's necessary to take into account the different

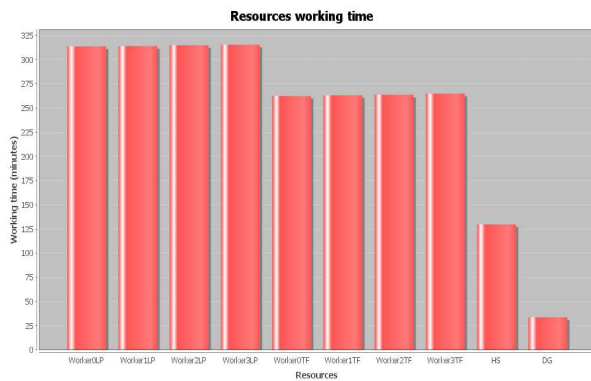


Figure 3. Resources working time.

kinds of process which are started per month. With that info it's possible to calculate the number of activities which are executed in that period of time. The activities are grouped in activities where is needed to generate any document or consult some info into a database and activities which execution is monotonous and the time spent is few. The first group had a bigger execution time than the second one. This assumption generates an equation with two unknown values. Using different pair of values it's possible to approximate the activities' execution time. The pair which emulates the system behavior in a better way on the simulation is used.

4.2.3. Flow modeling

The workflow of the process was presented in the Figure 1. The activities nodes are modeled yet so only the decision points need to be modeled. These points can be modeled using the patterns defined by Aalst in (Aalst et al. 2003). Using the ExclusiveChoice pattern with associated conditions for each exit branch, the Parallel pattern and the Synchronization pattern, the decision points the problem is resolved.

4.2.4. Elements of the flow

There are sixteen requests that customers can provide to start the Registry of Associations process. Each request represents a subprocess that generates a specific



Figure 4. Request in group 1 execution time.

workflow cover. These cover depending on several characteristics: if the subprocess documentation needs to be corrected or consulted, if the process resolution is negative or positive, etc. All of this variability is modeled using variables associated to each subprocess. These variables will be checked in the decision points which determinate the workflow path. The variables are initialized before the subprocess starts its execution. This initialization is associated to different probabilities which are obtained from the meeting with the government responsible. The frequency of the request generations are checked in the company's database. With that info is can be possible to determinate the statistic distributions of each request. These distributions are incorporated to the model.

All of the assumptions exposed before permits to generate a simulation model which can be executed by SIGHOS and obtain results of the Registry of Associations process behavior.

4.3. Registry of Associations simulation results

The simulation of the Registry of Associations process shows the behavior of the process. The Figure 2 shows the request started, solved and the request which are still on execution each month. This figure shows that there are picks of request requirements in a period of time every year. These picks generate a growth of the request execution queue. This growth stresses the system to levels that only can be recuperated several months after. In general where a period with a pick of request appears the resources can't assimilate all the work. This remaining work is assumed in periods where the request frequency is low.

The Figure 3 shows the working time of each resource in minutes. In case of Las Palmas' workers are more stressed that Santa Cruz's workers. This is possible because the request frequency is different in each province. The working time of the rest of resources isn't contributing important info. The amount of activities that are executed by these resources is small in comparison with the total amount of activities. The importance of this figure is to quantify the amount of time which all the resources are working with the actual system to compare it with the same value in the simulation of the electronic process.

Another simulation result that it's important to quantify is the average execution time for each user request. This average it's important to vitalize the possible benefits of an electronic process implementation. The user request can be grouped in three different types. One group contains request which need resolution and inscription. These request cover all the workflow so its execution time are bigger that the others. The second group contains request which only needs inscription and the third group contains request which don't need resolution and inscription. The Figure 4 shows the average time of the first group of user request.



Figure 5. User request graph with electronic processes.

4.4. Electronic processes application

The Canary Islands government contracts a company to develop tools and web services to increase the Registry of Associations productivity. This development will change the structure of process and as result, the behavior change too. After several meetings with the company the structural process changes to incorporate in the model were:

- The company estimates that its tools will make some activities easier than nowadays. These activities are: request registration, correction request send, resolution inscription, waiver development, process' inscription registration, correction request development and notification development.
- Web services allow that the head of service and the general director can sign documents of each province at any time. It won't be necessary to stay physically in one of the provinces to sign its documents.

The new simulation model for the new Registry of Associations process won't be too different than the older. Resources, activities, customers request and workflow are the same. The only point where the models differ is the activities' execution time. In the older model, workers activities are grouped in activities that generates documents or database consults and activities that generates monotonous work and spent less time than the first group. The company information shows that the monotonous activities will spend less execution time using the new tools. In the model, it will be specified by an execution time reduction. The execution is reduced by 50%.

New tools make some documents generation easier too. So the first group of activities is subdivided in two: one group which keep the actual execution time and other group that incorporate the new tool benefits. The new group's activities will spend less execution time than activities of the older group but more than the monotonous activities.

In case of the sign activities, the execution time represents the wait time to sign any document in the older model. The new web service removes wait queues

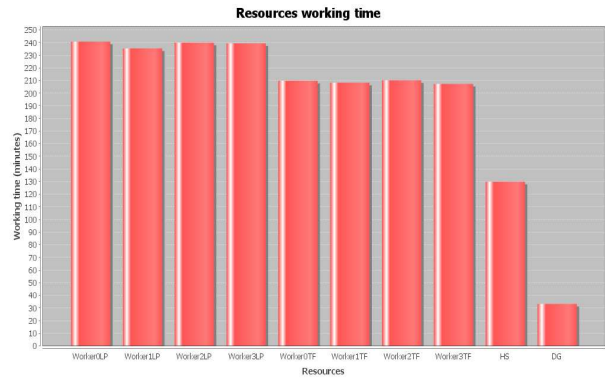


Figure 6. Resources working time with electronic processes.

so the wait time is not considerable now. In the new model the execution time is reduced to the minimum simulation time.

The changes adopted to implement the new model define the behavior process change. This change will be represented in the simulation results. The comparison between actual and new process simulation results quantifies the benefits or the loss of the electronic processes application.

4.5. Electronic processes application simulation

The simulation results show a process which can assume the user request generation without stress. Figure 5 shows that the request execution queue keeps its size in a low value during the months. This fact is refuted by the Figure 6. This figure shows the resources working time. The resources which assume a great part of the activities execution, workers, have long periods of inactivity. The average working time for Las Palmas province workers is around 240 minutes. It means that workers only need to work four hours to do the entire job.

Comparing these results with the actual system simulation results the benefits are obvious. In Figures 2 and 5 it is clear that the process stress reduction is substantial. The system can assume the request generation peaks in a best way. The execution queue size doesn't increase to high levels and keep a continuous size all the time. Focusing in the resources working time, the reduction is clear too as seen in

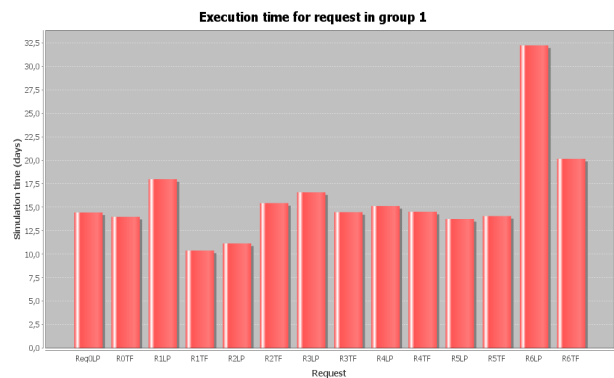


Figure 7. Request in group 1 execution time with electronic processes.

Figures 3 and 6. Workers of both provinces need less time to do the entire job.

In case of user benefits, the Figure 7 shows the execution time of some requests with the electronic process application. If data showed in Figure 4 and 7 are compared, the average of time benefits it's around 50%. So it's clear that low stress levels in the process' resources causes less waiting time for a request finalization.

In conclusion, a general review of the simulation results shows that the new tools and technology incorporation will cause several benefits for Canary Islands government and Registry of Associations customers. In case of Canary Island government, the actual request volume execution cost is less with the new implementations. In addition, workers can increase their productivity and face up a possible increase of the user request because of the electronic developments. In case of customers, they have to wait less time to obtain request responses by the Registry. It will cause a better level of satisfaction with the service.

5. CONCLUSIONS

In this paper, a simulation is developed to evaluate the incorporation of ICT technologies in a Canary Islands government process. The old process of Registry of Associations was compared with the new one using the simulation tool SIGHOS.

The results of the simulations reveal drawbacks in the actual process. The main problem is that the process' staff has to do a big effort to resolve all the process' inputs. This fact causes high levels of staff working time and in the customers waiting time.

The new process proposed tries to resolve these problems. The simulations have shown that new development can, theoretically, reduce the staff working time. Therefore Registry of Associations process could take on a higher volume of inputs than the actual process. Focusing in the customer waiting time, the new development causes a significant time reduction. This fact will increase the satisfaction customer level.

This paper proves that simulation can be used to implement eGovernment processes. Every change in an organization has to be justified and its benefits demonstrated. Simulation makes this possible. With this case, it has been confirmed that the applied simulation of business process as it provides insights into policies, practices, procedures and process flows.

It can be concluded that simulations have also demonstrated their usefulness in stochastic processes that are fully dependent on human stochastic behavior.

ACKNOWLEDGMENTS

This work is being supported by a project (reference DPI2006-01803) from the Ministry of Science and Technology with FEDER funds.

REFERENCES

- Aalst, W. M., Hofstede, A. H., Kiepuszewski, B., and Barros, A. P. (2003) "Workflow Patterns." *Distributed and Parallel Databases*, 14(1).
- Aguilar Chinae, R. M., Castilla Rodríguez, I., and Muñoz González, R. C. (2009) "Hospital Resource Management" in *Simulation-Based Case Studies in Logistics.*, pp. 65-84.
- Aguilar, R. M., Castilla, I., Muñoz, R., Martín, C. A., and Piñero, J. D. (2006) "Verification and Validation in Discrete Event Simulation: A Case Study in Hospital Management"
- Albores, P., Ball, P. D., and MacBryde, J. (2002) "Assessing the impact of electronic commerce in business processes: a simulation approach" in *Aalborg*, pp. 15-26.
- Baquero, P., Aguilar, R., Castilla, I., and Ayala, A. (2005) "Simulation of an IT Center for Users Attention" in *Marsella-Francia*.
- Greasley, A. (2005) "Using system dynamics in a discrete-event simulation study of a manufacturing plant." *International Journal of Operations & Production Management*, 25(6), pp. 534-548.
- Janssen, M. and Cresswell, A. (2005) "Enterprise Architecture Integration in E-Government." *HICSS*.
- Janssen, W., Steen, M., and Franken, H. (2003) "Business process engineering versus e-business engineering: a summary of case experiences." *Proceedings of the 36th Annual Hawaii International Conference on System Sciences*, 2003..
- Kovacic, A. and Pecek, B. (2007) "Use of Simulation in a Public Administration Process." *Simulation*, 83(12), p. 10.
- Kumar, S. and Phrommathed, P. (2006) "Improving a manufacturing process by mapping and simulation of critical operations." *Journal of Manufacturing Technology*, 17(1), pp. 104-132.
- Li, Z. (2008) "How E-government affects the organisational structure of Chinese government." *AI & Society*, 23(1), p. 7.
- Moon, Y. and Phatak, D. (2005) "Enhancing ERP system's functionality with discrete event simulation." *Industrial Management & Data Systems*, 105(9), pp. 1206-1224.
- Nikolaidou, M., Anagnostopoulos, D., and Tsalgaidou, A. (2001) "Business process modeling and automation in the banking sector: A case study.." *International Journal of Simulation Systems, Science & Technology, Special Issue on: Business Process Modeling*, 2(2), pp. 65-76.
- Ozbay, K. and Bartin, B. (2003) "Incident Management Simulation." *Simulation*, 79(2), pp. 69-82.
- Scholl, H. (2003) "E-government: a special case of ICT-enabled business process change." *Proceedings of the 36th Annual Hawaii International Conference on System Sciences*, 2003.

Wu, J. and Hu, B. (2007)" Modeling and simulation of group behavior in e-government implementation." Winter Simulation Conference, p. 7.

AUTHORS BIOGRAPHY

YERAY CALLERO was born in Haría, Lanzarote and attended the University of La Laguna, where he studied Engineering Computer Science and obtained his degree in 2008. He is currently working on his PhD with the Department of Systems Engineering and Automation at the same university. His research interests include simulation and embedded system software design.

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. Her current research interests are decision making based on discrete event simulation systems and knowledge-based systems, intelligent agents, and intelligent tutorial systems.

USING DISCRETE-EVENT SIMULATION TO DESIGN RELIABLE AND COST-EFFICIENT CIVIL-ENGINEERING STRUCTURES

A. Juan ^(a), A. Ferrer ^(b), C. Serrat ^(b), J. Faulin ^(c), J. Hester ^(a), P. Lopez ^(b)

^(a) Dep. of Computer Science, Multimedia and Telecommunication – IN3, Open University of Catalonia, Spain

^(b) School of Building Construction of Barcelona – IEMAE, Technical University of Catalonia, Spain

^(c) Dep. of Statistics and IO, Public University of Navarre, Spain

^(a) ajuana@uoc.edu, jchester@mit.edu, ^(b) alberto.ferrer@upc.edu, carles.serrat@upc.edu, pere.lopez@upc.edu,

^(c) javier.faulin@unavarra.es

ABSTRACT

In this paper the topic of Structural Reliability Analysis and its importance in modern Civil Engineering is introduced. Throughout the paper, some advantages that simulation-based approaches offer with respect to other classical approaches are discussed. After a review of the most relevant literature on the subject, a simulation-based approach is described. Our methodology makes use of discrete-event simulation techniques to help the civil engineer design more reliable and cost-efficient structures. A numerical example illustrates some potential applications of the proposed methodology. Finally, potential research and academic applications of this approach are also discussed.

Keywords: structural reliability, discrete-event simulation

1. INTRODUCTION

As any other physical system, buildings and civil engineering structures suffer from age-related degradation due to deterioration, fatigue and deformation. These structures also suffer from the effect of external factors like corrosion, overloading, environmental hazards, etc. Thus, the state of any structure should not be considered to be constant as often happens in structural literature, but rather as being variable through time. For instance, reinforced concrete structures are frequently subject to the effect of aggressive environments (Stewart and Rosowsky 1998). According to Li (1995) there are three major ways in which structural concrete may deteriorate, namely: (i) surface deterioration of the concrete, (ii) internal degradation of the concrete and (iii) corrosion of reinforcing steel in concrete. Of these, reinforcing steel corrosion is the most common and is the main target for the durability requirements prescribed in most design codes for concrete structures. In other words, these structures suffer from different degrees of resistance deterioration due to aggressive environments and, therefore, reliability problems associated with these structures should always consider the time dimension.

For any given structure, it is possible to define a set of limit states (Melchers 1999). Violation of any of these limit states can be considered a structural failure of a particular magnitude, type or level, and represents an undesirable condition for the structure. In this sense, Structural Reliability is an engineering discipline that provides a series of concepts, methods and tools to predict and/or determine the reliability, availability and safety of buildings, bridges, industrial plants, off-shore platforms and other structures, both during their design stage and during their useful life. Structural Reliability should be understood as the structure's ability to satisfy its design goals for some specified time period. From a formal perspective, Structural Reliability can be defined as the probability that a structure will not achieve a specified limit state –i.e. will not suffer a failure of a certain type– during a specified period of time (Thoft-Christensen and Murotsu 1986). For each identified failure mode, the failure probability of a structure is a function of operating time, t , and it may be expressed in terms of the distribution function, $F(t)$, depending on the time-to-failure random variable, T . The reliability or survival function, $R(t)$, which is the probability that the structure will not have achieved the corresponding limit state at time $t > 0$, is then given by $R(t) = 1 - F(t) = P(T > t)$. According to Petryna and Krätzig (2005), interest in structural reliability analysis has been increasing in recent years, and today it can be considered a primary issue in civil engineering. From a reliability point of view, one of the main targets of structural reliability is to provide an assembly of components that, when acting together, will perform satisfactorily –i.e., without suffering from critical or relevant failures– for some specified time period, either with or without maintenance policies.

In this article we discuss the use of discrete-event simulation (DES) as the most natural way to deal with uncertainties in time-dependent structural reliability. After reviewing some previous works that promote the use of simulation-techniques –mainly Monte Carlo simulation– in the structural reliability arena, we discuss how DES can be employed to offer solutions to structural reliability and availability problems in

complex scenarios. We illustrate some potential applications of our methodology with a numerical case study. Finally, we discuss some advanced topics for further research and highlight potential academic applications of our approach.

2. RELATED WORK

As noticed by Lertwongkornkit et al. (2001), it is becoming increasingly common to design buildings and other civil infrastructure systems with an underlying “performance-based” objective, which might consider more than just two structural states (collapsed or not collapsed). This makes it necessary to use techniques other than just design codes in order to account for uncertainty in key random variables affecting structural behavior. According to other authors (Marek et al. 1996) standards for structural design are basically a summary of the current “state of knowledge”, but they only offer limited information about the real evolution of the structure through time. So they strongly recommend the use of probabilistic techniques as a more realistic alternative.

As Park et al. (2004) point out, it is difficult to calculate probabilities for each limit-state of a structural system. Structural reliability analysis can be performed using analytical methods or simulation-based methods. On one hand, analytical methods tend to be complex and generally involve restrictive simplifying assumptions on structural behavior, which makes them difficult to apply in real scenarios. On the other hand, simulation-based methods allow incorporating realistic structural behavior (Laumakis and Harlow 2002). Traditionally, simulation-based methods have been considered to be computationally expensive, especially when dealing with highly reliable structures (Marquez et al. 2005). This is because when there is a low failure rate, a large number of simulations is needed in order to get accurate estimates. Nevertheless, in our opinion these computational concerns can be now considered mostly obsolete due to the outstanding improvement in processing power experienced in recent years.

Monte Carlo simulation has often been used to estimate the probability of failure and to verify the results of other reliability analysis methods. In this technique, the random loads and random resistance of a structure are simulated, and these simulated data are then used to find out if the structure fails or not according to pre-determined limit states. The probability of failure is the relative ratio between the number of failure occurrences and the total number of simulations. Monte Carlo simulation has been applied in structural reliability analysis for at least three decades now. Kamal and Ayyub (1999) were probably the first to use DES for reliability assessment of structural systems that would account for correlation among failure modes and component failures. Recently, Song and Kang (2009) presented a numerical method based on subset simulation to analyze structural reliability. Our approach is inspired by previous work on system

reliability and availability developed by Juan et al. (2007, 2008) and Faulin et al. (2007, 2008).

3. THE LOGIC BEHIND OUR APPROACH

Consider a structure with several components. These components are connected together according to a well-defined logical topology. Assume also that time-dependent reliability functions for each component are known. This information might have been obtained from historical records or, alternatively, from survival analysis techniques –e.g. accelerated-life tests– on individual components. Therefore, at any moment in time the structure will be in one of the following states: (a) perfect condition, i.e.: all components are in perfect condition and thus the structure is fully operational; (b) slight damage, i.e.: some components have experienced failures but this has not affected the structural operability in a significant way; (c) severe damage, i.e.: some components have failed and this has significantly limited the structural operability; and (d) collapsed, i.e.: some components have failed and this might imply structural collapse. Notice that, under these circumstances, there are three possible types of structural failures depending upon the state that the structure has reached. Of course, the most relevant – and hopefully least frequent– of these structural failures is structural collapse, but sometimes it might also be interesting to be able to estimate the reliability or survival functions associated with other structural failures as well. To achieve this goal, DES can be used to artificially generate a random sample of structural lifecycles (Figure 1).

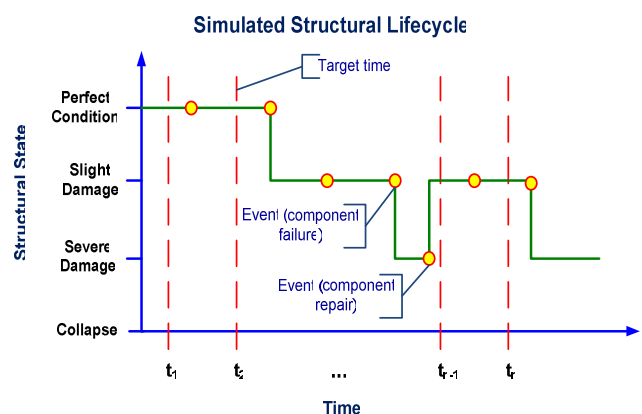


Figure 1: Generating structural lifecycles with DES

For each of these randomly generated lifecycles, an observation of the structural state is obtained at each target-time. After running a large number of iterations –e.g. some hundred thousands or millions–, accurate point and interval estimates can be calculated for the structural reliability at each target-time (Juan et al. 2007). Also, additional information can be obtained from these runs, such as: which components are more likely to fail, which component failures are more likely to cause structural failures (failure criticality indices), which structural failures occur more frequently, etc. Moreover, notice that DES could also be employed to

analyze different scenarios (what-if analysis), i.e.: to study the effects of a different logical topology on structural reliability, the effects of adding some redundant components on structural reliability, or even the effects of improving reliability of some individual components. Finally, DES also allows for considering the effect of maintenance policies or dependencies among component failures (Faulin et al. 2008).

4. A SIMULATION-BASED METHODOLOGY

Figure 2 summarizes the methodology employed in our approach:

1. Given a structural design –or the corresponding information for an already constructed structure–, the first step is to obtain reliability information, i.e., the statistical distributions that model the time-to-failure random variable, for each component. As stated before, this can either be done by using historical data or by doing accelerated-life tests in a laboratory, so we will consider that this information is already available to us.
2. Then, we must be able to identify the different structural failure modes. Notice that the concept of structural failure can be user-defined. In general, it is possible to consider a different type of failure for each limit state that a structure can reach. Sometimes we can be interested in analyzing a particular type of failure. For example, we could consider that the structure has failed whenever it reaches a state of severe damage, even when it has not yet collapsed. By identifying all structural failure modes, we can construct the logical topology of the structure. That is, we can list which combinations of component failures will bring the structure to the limit state under study. Obtaining this logical topology will not always be a trivial task, and some special techniques might be necessary (Faulin et al. 2007).
3. Using the failure-time distributions for each component, a simulation can be run where discrete events are generated to represent failures of different structural components. This, in turn, allows emulating the lifecycle of a structure, which provides information regarding the state of the structure at different target-times. At each of these target-times, the state of the structure can be calculated by combining the current state of each component with the logical topology previously discussed.
4. After generating some thousands or millions of random structural lifecycles, a large set of random observations of the structure’s status at each target time is available. From these observations, it is possible to obtain –by using statistical inference– interval estimates for the structural reliability function, i.e.: the probability of the structure not having failed yet at any given target-time.
5. Moreover, other relevant information can also be derived from the simulations being performed. For instance, it is possible to identify those components

that are especially critical from a structural reliability point of view. This is obtained by calculating failure criticality indices like the percentage of times that each component failure is causing an overall structural failure.

6. Of course, both the information on the structural reliability function and the failure criticality indices can be used to improve the structural design so that the resulting structure will be more reliable through time.

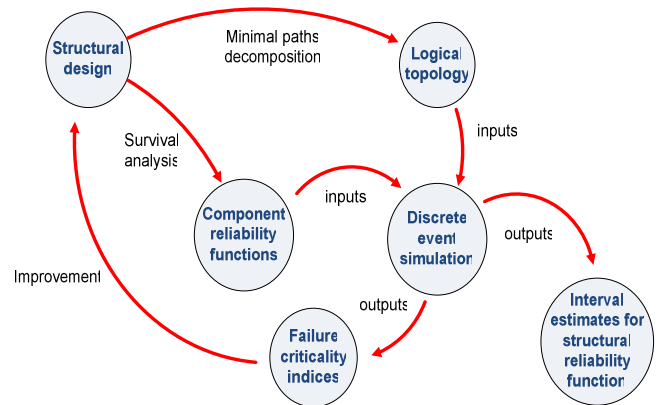


Figure 2: Scheme of our approach

5. A NUMERICAL EXAMPLE

We present here a case study of three possible designs for a bridge, which is inspired by a previous example from Saka (1990). As can be seen in Figure 3, there is an original design (case A) and two different alternatives, one with redundant components (case B) and another with reinforced components (case C).

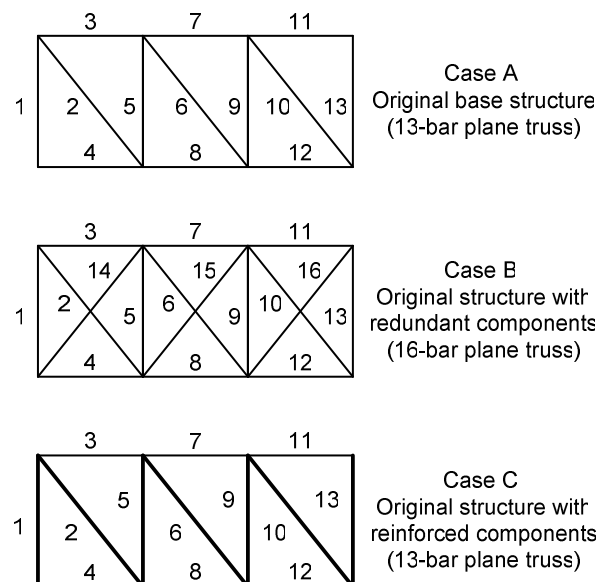


Figure 3: Different possible designs for a structure

Our goal is to illustrate how our approach can be used in the design phase to help pick the most appropriate

design, depending on factors such as the desired structural reliability, the available budget (cost factor) and other project restrictions. As explained before, different levels of failure can be defined for each structure, and in examining how and when the structures fail in these ways, one can measure their reliability as a function of time. Different survival functions can be then obtained for a given structure, one for each structural failure type. By comparing the reliability of one bridge to another, one can determine whether a certain increase in structural robustness – either via redundancy or via reinforcement– is worthwhile according to the engineer’s utility function. As can be deduced from Figure 3, the three possible bridges are the same length and height, but the second one (case B) has 3 more trusses connecting the top and bottom beam and is thus more structurally redundant. If the trusses have the same dimensions, the second bridge should have higher reliability than the first one (case A) for a longer period of time. Regardless of how failure is defined for the first bridge, a similar failure should take longer to occur in the second bridge. Analogously, the third bridge design (case C) is likely to be more reliable than the first one (case A), since it uses reinforced components with improved individual reliability (in particular, components 1’, 2’, 5’, 6’, 9’, 10’ and 13’ are more reliable than their corresponding components in case A).

Let us consider three different types of failure. Type 1 failure corresponds to slight damage, where the structure is no longer as robust as it was at the beginning but it can still be expected to perform the function it was built for. Type 2 failure corresponds to severe damage, where the structure is no longer stable but it is still standing. Finally, type 3 failure corresponds to complete structural failure, or collapse. Now we have four states to describe the structure, but only two (failed or not failed) to describe each component of the structure. We can track the state of the structure by tracking the states of its components. Also, we can compare the reliabilities of the three different structures over time, taking into account that different numbers of component failures will correspond to each type of structural failure depending on the structure. For example, a failure of one component in the case A and C bridges could lead – being conservative– to a type 3 failure, while it will only lead to a type 2 failure in the case B bridge. In other words, for case B it will take at least two components to fail in the same section of the bridge before the structure experiences a type 3 failure.

In order to develop a numerical example, we assumed that the failure-time distributions associated with each individual truss are known. Table 1 shows these distributions. As explained before, this is a reasonable assumption since this information can be obtained either from historical data or from accelerated-life tests.

Table 1: Failure-times at component level

Failure-times for each of the trusses *							
N	D	Shape	Scale	N	D	Shape	Scale
1	W	15	50	9	W	15	50
1'	W	18	55	9'	W	18	55
2	W	13	45	10	W	13	45
2'	W	17	60	10'	W	17	60
3	W	20	55	11	W	20	55
4	W	22	55	12	W	22	55
5	W	15	50	13	W	15	50
5'	W	18	55	13'	W	18	55
6	W	13	45	14	W	13	45
6'	W	17	60	15	W	13	45
7	W	20	55	16	W	13	45
8	W	22	55				

(*) D means distribution while W means Weibull

To numerically solve this case study we used the SURESIM software application (Juan et al. 2008), which implements the algorithms described in our methodology. We ran the experiments in a standard PC, Intel Pentium 4 CPU 2.8GHz and 2GB RAM. Each case was run for one million iterations, each iteration representing a structural life-cycle for a total of 1E6 observations. The total computational time employed for running all iterations was below 10 seconds for each test, which demonstrates the computational feasibility of our approach. Figure 4 shows the survival or reliability functions obtained for each case. This survival function shows the probability that each bridge will not have failed after some time (expressed in years). As expected, both cases B and C represent more reliable structures than case A. In this example, case C (reinforced components) shows itself to be an even more reliable design than case B (redundant components). Notice that this conclusion holds only for the current values in Table 1. That is, should the shape and scale parameters change, case B could then be more reliable than case C.

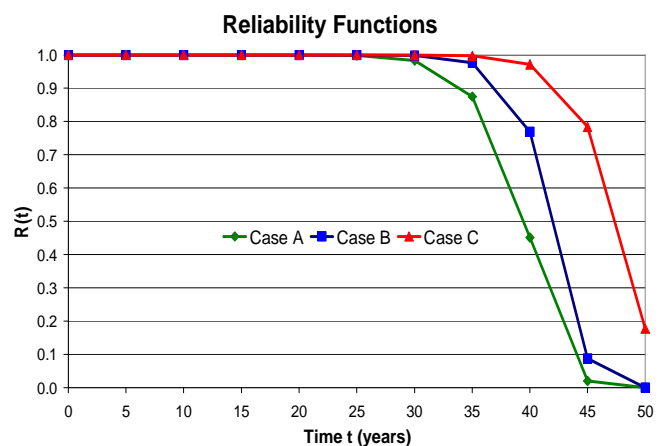


Figure 3: Different possible designs for a structure

Table 2 shows the expected or average life for each bridge design. Notice how, again, case C is the most reliable.

Table 2: Expected life for each bridge

Structural Mean Time To Failure	
Case	Years
A	39.13
B	41.70
C	47.17

Finally, Figure 4 shows failure criticality indices for each considered design. Notice that in case A the most critical components are trusses 2, 6 and 10. Since in case C these components –and others– have been reinforced, they do not appear as the critical components anymore. Also in case B their levels of criticality have been reduced somewhat due to the introduction of redundancies.

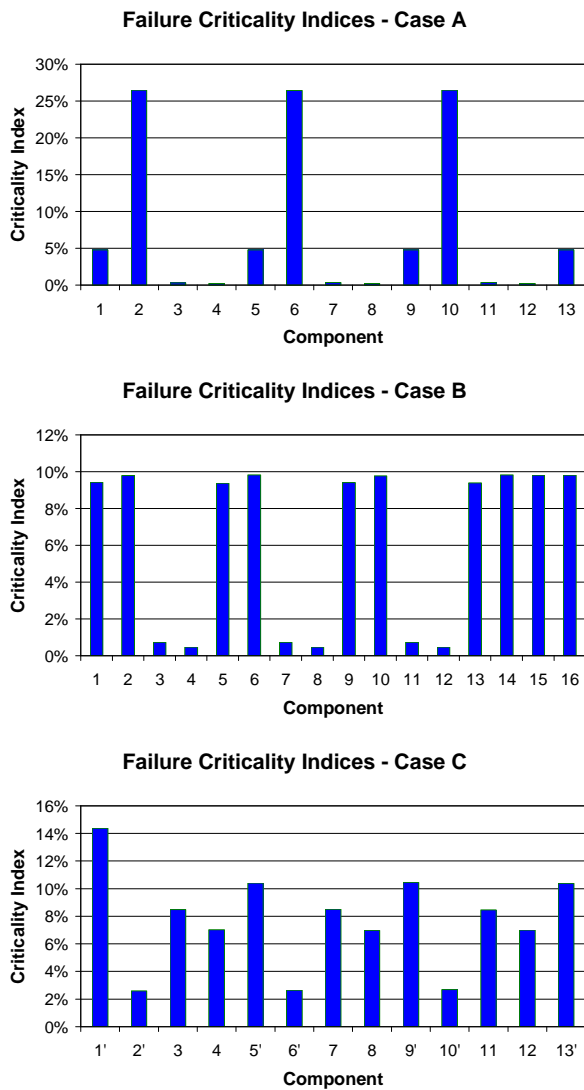


Figure 4: Failure Criticality Indices for each case

6. SOME ADVANCED TOPICS

Our simulation-based approach also allows for considering maintenance policies –i.e. component repairs– or even dependencies among component failures. In order to consider maintenance policies, repair-time distributions for each component should be known in advance. Then, component repairs will simply be introduced in the model as a new kind of event that will be randomly generated during the simulated structural lifecycle. Moreover, it is usually the case that a component failure can affect the failure rate of other components –i.e., component failure-times are not independent in most real situations. Again, discrete-event simulation can handle this complexity by simply updating the failure-time distributions of each component each time a new component failure takes place. This way, dependencies can be also introduced in the model. Notice that this represents a major difference between our approach and other approaches –mainly analytical ones–, where dependencies among components, repair-times or multi-state structures are difficult to consider.

7. ACADEMIC APPLICATIONS

The use of simulation as a methodology to solve structural reliability problems is not only interesting in the professional arena but also in the academic one (Lertwongkornkit et al. 2001). Because of technological improvements in both software and hardware, we can now run simulations of lengths and complexities that were once not possible. These new methods not only can deliver valid estimates of the structural reliabilities of complex structures, but they can also provide a new tool in the classroom to enhance students' comprehension of internal deterioration processes. Also, as discussed in this paper, students can use simulation to analyze alternative scenarios and designs (what-if analysis) for one structure.

8. CONCLUSIONS

In this paper, the convenience of using probabilistic methods to estimate reliability and availability in time-dependent building and civil engineering structures has been discussed. Among the available methods, discrete-event simulation seems to be the most realistic choice, especially during the design stage, since it allows for efficient comparison of different scenarios. Discrete-event simulation offers clear advantages over other approaches, namely: (a) the opportunity of creating models which faithfully reflect the real structure characteristics and behavior, and (b) the possibility of obtaining additional information about the system and its critical components.

ACKNOWLEDGMENTS

This work has been partially supported by the IN3-UOC Knowledge Community Program (HAROSA) and by the Institute of Statistics and Mathematics Applied to the Building Construction (IEMAE), which belongs to the School of Building Construction of Barcelona.

REFERENCES

- Faulin, J., Juan, A., Serrat, C. and Bargueño, V. 2007. Using Simulation to determine Reliability and Availability of Tele-communication Networks. *European Journal of Industrial Engineering* 1(2):131-151.
- Faulin, J., Juan, A., Serrat, C. and Bargueño, V., 2008. Improving Availability of Time-Dependent Complex Systems by using the SAEDES Simulation Algorithms. *Reliability Engineering and System Safety* 93(11):1761-1771.
- Juan, A., Faulin, J., Serrat, C., Sorroche, M. and Ferrer, A. 2008: A Simulation-based Algorithm to Predict Time-dependent Structural Reliability. *Advances in Simulation for Production and Logistics Applications*, pp. 555-564. Fraunhofer IRB, Stuttgart, Germany.
- Juan, A., Faulin, J., Sorroche, M. and Marques, J. 2007. J-SAEDES: A Simulation Software to improve Reliability and Availability of Computer Systems and Networks. *Proceedings of the 2007 Winter Simulation Conference* 2285-2292. Washington (Washington D.C., USA).
- Kamal, H. and Ayyub, B., 1999. Reliability Assessment of Structural Systems Using Discrete-Event Simulation. *Proceedings of the 13th ASCE Engineering Mechanics Division Specialty Conference*. Baltimore (MD, USA).
- Laumakis, P., and Harlow, G., 2002. Structural Reliability and Monte Carlo Simulation. *International Journal of Mathematical Education in Science and Technology* 33(3):377-387.
- Lertwongkornkit, P., Chung, H. and Manuel, L., 2001. The Use of Computer Applications for Teaching Structural Reliability. *Proceedings of the 2001 ASEE Gulf-Southwest Section Annual Conference*, Texas (Texas, USA).
- Li, C., 1995. Computation of the Failure Probability of Deteriorating Structural Systems. *Computers & Structures* 56(6):1073-1079.
- Marek P., Gustar M. and Anagnos, T., 1996. *Simulation Based Reliability Assessment for Structural Engineers*. CRC Press, Boca Raton, FL, USA.
- Marquez, A., Sanchez, A. and Iung, B., 2005. Monte Carlo-Based Assessment of System Availability. a Case Study for Cogeneration Plants. *Reliability Engineering & System Safety* 88(3):273-289.
- Melchers, R., 1999. *Structural Reliability: analysis and prediction*. John Wiley & Sons.
- Park, S., Choi, S., Sikorsky, C. and Stubbs, N., 2004. Efficient method for calculation of system reliability of a complex structure. *Int. J. of Solids and Structures* 41:5035-5050.
- Petryna, Y. and Krätzig, W., 2005. Computational framework for long-term reliability analysis of RC structures. *Comput. Methods Appl. Mech. Eng.* 194(12-16):1619-1639.
- Saka, M. P., 1990. Optimum design of pin-jointed steel structures with practical applications. *Journal of Structural Engineering-Asce* 116(10):2599-2620.
- Song, J. and Kang, W., 2009. System reliability and sensitivity under statistical dependence by matrix-based system reliability method. *Structural Safety* 31(2):148-156.
- Stewart, M., and Rosowsky, D., 1998. Time-dependent reliability of deteriorating reinforced concrete bridge decks. *Structural Safety* 20:91-109.
- Thoft-Christensen, P. and Murotsu, Y., 1986. *Application of Structural Systems Reliability Theory*. Springer Verlag.

AUTHORS BIOGRAPHY

Angel A. Juan is an Associate Professor of Simulation in the Computer Science Department at the Open University of Catalonia (UOC), Spain. He holds a Ph.D. in Industrial Engineering, an M.S. in Information Technologies, and a M.S. in Applied Mathematics. His research focuses on Simulation applications.

Albert Ferrer is an Associate Professor in the Department of Applied Mathematics I at the Technical University of Catalonia (UPC), Spain. He holds a Ph.D. in Mathematics and a M.S. degree in Computing. His research fields are Abstract Convex Analysis, Nonlinear Optimization, Global Optimization and Fuzzy Sets Theory.

Carles Serrat is an Associate Professor of Applied Statistics in the Department of Applied Mathematics I at the Technical University of Catalonia (UPC), Spain. He holds a PhD on Statistics and a BS on Applied Mathematics. He develops his research in the Institute of Statistics and Mathematics Applied to the Building Construction (IEMAE).

Javier Faulin is an Associate Professor of Operations Research at the Public University of Navarre, Spain. He holds a PhD in Economics, a MS in Operations Management and a MS in Applied Mathematics. His research interests include Logistics and Simulation. He is an editorial board member of the Int. J. of Applied Management Science.

Josh Hester is a Civil Engineering student at the Massachusetts Institute of Technology, USA. He is also a visiting student at the Open University of Catalonia under the IN3-KC program.

Pere Lopez is an Assistant Professor in the Department of Applied Mathematics I at the Technical University of Catalonia (UPC), Spain. He holds a M.S. degree on Civil Engineering and is a member of the Institute of Statistics and Mathematics Applied to the Building Construction (IEMAE).