MODELLING AND SIMULATION OF AUTONOMOUS LOGISTIC PROCESSES

Bernd Scholz-Reiter$^{(a)}$, Torsten Hildebrandt$^{(a)}$, Jan Kolditz $^{(a)}$

$^{(a)}$ Planning and Control of Production Systems, University of Bremen, Germany

$^{(a)}$ {bsr, hil, kol}@biba.uni-bremen.de

ABSTRACT

Autonomous control of logistic processes is proposed as a means for enterprises to better face dynamics and complexity, caused e.g. by globalization. This conference contribution will first briefly sketch the idea of autonomous control of logistic processes. Second it will discuss existing modelling approaches on the basis of requirements to a suitable method and subsequently will outline our modelling method designed for engineering of autonomous logistic processes. The third part will detail the transformation of conceptual models constructed by using this method into a simulation on the basis of an industrial case study. Moreover some results of simulation studies will be presented and discussed. The paper is concluded by a summary and an outlook on future work.

Keywords: autonomous control, process modelling,
agent oriented simulation, modelling method

1. INTRODUCTION

Coping with complexity of logistic systems is a task undertaken only insufficiently today. One reason is that the centralised planning and control strategies used presume information that is mostly not available in the required quality and quantity. Autonomous control within the context of the German interdisciplinary research effort CRC 637 means processes of decentralized decision making of interacting system elements in heterarchical structures. Concretised towards autonomous control of logistic processes it is defined as "[…] characterized by the ability of logistic objects to process information, to render and to execute decisions on their own" (Huelsmann and Windt 2007). A logistic object fulfilling this definition is called an autonomous logistic object; to support its design implicates an approach focused on these objects. RFID (radio frequency identification)- and smart label-technologies and their successors in the foreseeable future are seen as an enabling technology to realize autonomous control of logistic processes.

The design of autonomous logistic processes is an extensive task, whose complexity normally impedes its complete notional analysis and design. Therefore it is necessary to utilise construction of models where in particular graphical methods allow descriptive and easily comprehensible models. For supporting such a modelling task on one hand methods from business process modelling and on the other hand methods for software agent modelling should be considered.

In section 2 this paper sketches the engineering of a system based on autonomous logistic processes and discuss different existing modelling approaches on the basis of our specific requirements in that context. In section 3 we outline a modelling method specifically designed for conceptual modelling during the engineering phase of systems based on autonomous logistic processes.

Section 4 presents an industrial case study transforming the conceptual model of a manufacturer of automotive supplies into a simulation model. The subsequent sections 5 and 6 present results from a simulation study.

2. MODELLING IN THE CONTEXT OF ENGINEERING SYSTEMS BASED ON AUTONOMOUS LOGISTIC PROCESSES

2.1. Engineering autonomous logistic processes and requirements on an adequate modelling method

On the basis of the general Systems Engineering procedure model (Haberfellner 2002) the engineering of an autonomous logistic system can be described by the phases initiation, preliminary study, specification, simulation, infrastructure configuration, cost benefit estimation, establishment and introduction (Scholz-Reiter et al. 2007). The four phases from specification to cost benefit estimation form the methodical core as an iterative process.

In the specification phase a conceptual model of the system is created in the form of a semi-formal specification of the autonomous logistic objects. Moreover identification, design and allocation of decision processes are performed. It has to be clarified which elements are part of the system and which of them are "intelligent" respectively autonomous entities.

During the simulation phase the design created before is tested. Especially operability and impact on logistics performance of the whole system are focused here. This step therefore allows an effective comparison to the existing or alternative ways of controlling the examined logistic system. The simulation code may already be part of the engineering process of the planned control software if the code is reusable. Otherwise the core software engineering process starts in the implementation phase.

On the basis of the insights gained before an estimation of needed hardware equipment for the autonomous system (for example what kind of communication infrastructure) can be made, getting more detailed with every iteration loop. For example from allocation of control processes and data packets to entities of the logistic system necessary memory and computing capacity may be derived.

Every iteration is concluded by a cost benefit estimation. On the basis of the rating and subsequent decision the original process model can be adjusted according to the new conclusions. In case of repeated negative results in this step an application of autonomous control has to be abandoned for this scenario.

Using the definition of autonomous logistic processes and the necessary phases for engineering a system based on this principle four main requirements can be formulated on a methodology for modelling autonomous logistic processes.

- The methodology has to fulfil the general definitional attributes of a modelling methodology. Therefore it must imply at least a notation, a procedure model and a fundamental structuring like a view concept (definition orientation).
- The methodology must be application area oriented and therefore has to aim first at planning and control of logistic systems and second at modelling the constitutive characteristics of autonomy in logistics (application area orientation).
- The methodology must be appropriate for the user, for which reason an explicit consideration of logistic domain experts has to be assured (user orientation).
- The methodology must consider the use of models created with it, what especially requires consideration of subsequent agent oriented software implementation (model usage orientation).

2.2. Evaluation of existing approaches

In this subsection existing approaches for agent oriented modelling as well as process and logistic oriented modelling are examined concerning their appropriateness for modelling autonomous logistic processes. For the evaluation especially the four main requirements formulated before are relevant. In table 1 the evaluation is summarised by comparing methods and requirements. In case of a positive rating (+) the requirement is predominantly fulfilled, in case of a neutral rating (0) the requirement is partly fulfilled and in case of a negative rating (-) the requirement is inadequately fulfilled.

Regarding the software implementation the concept of agent-oriented software engineering is very close to the paradigm of autonomy in logistics due to the attributes of a software agent (Wooldridge and Jennings 1995) like autonomy, reactivity or adaptivity. Important agent oriented methods are Gaia (Cernuzzi et al. 2004), MaSE (DeLoach et al. 2004), MAS-CommonKADS (Iglesias et al. 1998), Tropos (Bresciani et al. 2004) and DACS (Bussmann et al. 2004). However in spite of the numerous existing methodologies for agent oriented software engineering, the deficits in connecting the software engineering with real production systems or with industrial systems in general is seen as one cause for the relatively low number of agent based systems actually used in industry (Monostori et al. 2006), (Hall et al. 2005). For Holonic Manufacturing Systems (HMS) (Valckenaers et al. 1999), which can be seen as an important approach to autonomy (Windt 2006), a significant demand for methods based on software engineering principles is seen, which support the designer of the HMS software system in all stages of the development process (Giret and Botti 2005), (McFarlance and Bussmann 2003). A main aspect of the insufficient methodical support is the requirements analysis and thus the linkage between real scenario and HMS-based software system (Giret and Botti 2006). In general agent-oriented software engineering methodologies accentuate important aspects like autonomy but widely disregard the decisions (Bussmann et al. 2004) being a constitutive characteristic of autonomous logistics processes. Moreover according to their intended use they focus on a detailed design of a software system but disregard the integration of a logistics domain expert in the specification of the system.

Table 1: Comparing Requirements and Existing Modelling Approaches

	ARIS	CIMOSA	IUM	SOM	MPSF	Gaia	MaSE	MAS-Common KADS	Tropos	DACS
definition orientation	+	+	+	+	0	+	+	+	+	0
application area orientation	-	-	-	-	0	0	0	0	0	0
user orientation	+	0	+	+	+	-	-	-	-	+
model usage orientation	0	0	0	0	0	+	+	+	+	+

The lacking consideration of domain experts does not apply to the Methodology for Designing Agent Based Production Control Systems (DACS) (Bussmann et al. 2004). DACS is meant to support a production engineer without experience in software agent technology during design of an agent based production control system.

In DACS no standardised notation is used but so-called trigger diagrams. These easily get very complex and unclear and are no longer presentable in a single diagram, but there are no possibilities mentioned to decompose. The central step of the DACS procedure is the agent identification. In most cases this step should lead to an aggregation of decisions in agents that represent physical components of the production system. That assignment of agents to physical elements is also a result of the case study and is endorsed by the used principle to identify things rather than function (Parunak et al. 1998). But this puts the extensive process of decomposing and composing of decision tasks into question. Rather using the physical components as orientation from the beginning seems to be more adequate. Furthermore it is not evident that the final definition of the agents to be used should be made by a production engineer who is not experienced in agent technology at all. For example it may be necessary to split up tasks because of limited functions of a single agent and therefore to partly abandon the physical component orientation. Nevertheless this orientation is reasonable especially during conceptual design even if more agents are responsible for the tasks of a single physical component. Moreover DACS does not support an iterative process, what would be necessary for the design of a system with realistic complexity.

Important process modelling methods are ARIS (Scheer 2000), CIMOSA (Vernadat 2006), IUM (Mertins and Jaekel 2006), SOM (Ferstl and Sinz 2006) and MPSF (Dangelmaier 2001). In context of software engineering, methodologies for business process modelling are intended to support the development of centralised information systems (Scheer 2000). Because of this purpose dedicated concepts for specifying decentralised approaches are missing. In principal these can be included by adjusting existing reference models (Boese and Windt 2007), but still there is a lack of sufficient instruments for explicit modelling of communication processes and protocols. However the most important aspect is that there is no guidance for designing a logistic system under consideration of autonomic control strategies and therefore no dedicated procedure model exists. On one hand methodologies for process modelling like ARIS or IUM do not take special care about design of planning and control processes. On the other hand the concept for model-based planning and control of production systems (MPSF) aims at the design of planning and control processes, but there a strict hierarchical and centralised approach is pursued.

According to these aspects the modelling methodology in context of engineering autonomous logistic systems shall be the connection between real-world oriented business process modelling and agent-oriented software engineering for the specific domain. The specification should focus on the planning and control processes of the real system or the system to be realised respectively. However the constructed model shall to some extend still be independent from the detailed software design. For example the logistic objects in an autonomous logistic system like machines, commodities or conveyors may be modelled as single autonomous entities, but the software architecture may differ. This flexibility allows the software engineer to split up abilities of a logistic object on multiple software agents when this is required because of favoured agent architectures or practical limits of a single agent. These activities of specifying the control processes on one hand and of designing the software system on the other hand require different qualifications. Thus a software engineer is in charge of the software design and therefore the determination of the software agent architecture. In contrast a logistics domain expert specifying the autonomous logistic system is responsible for planning and control processes and constructs a model that formulates requirements to the software system. When several people with different qualifications are involved in engineering a system, a modelling notation that is persistently used from the process model of the system to the implementation of the software avoids a gap in the engineering process by using standardised semantic concepts in the different disciplines (Specker 2005). One possibility for this is the use of the Unified Modeling Language (UML). As a graphical, semi-formal notation it is broadly used - besides software development (especially agent-oriented approaches are of particular interest here, see for instance AUML (Odell et al. 2001, Bauer et al. 2001) it is also used for knowledge modelling (Schreiber et al. 2001) or business process modelling (Oestereich et al. 2003).

3. METHOD FOR CONCEPTUAL MODELLING OF AUTONOMOUS LOGISTIC PROCESSES

In this section the first phase of the modelling cycle, the semi-formal specification of the autonomous logistic system, is focused. To support this specification a modelling methodology as part of the Autonomous Logistics Engineering Methodology (ALEM), with its components ALEM-N (ALEM-Notation), ALEM-P (ALEM-Procedure) and ALEM-T (ALEM-Tool), is proposed. ALEM-N consists of a view concept comprised of views each showing specific aspects of the logistic system as well as the notational elements to be used in each view and their intended meaning. ALEM-P is a procedure model describing the steps to be followed in generating a model and is intended to guide the user through analysis and specification of an autonomously controlled logistic system. ALEM-T is a software tool, specifically tailored to support the notation and the procedure model.

3.1. View Concept and Notation

Creating process models usually leads to a high degree of complexity. A view concept serves as a means to reduce the complexity constructing a model (Scheer 2000). ALEM-N includes a view concept for modelling of autonomous logistic processes, distinguishing five different views. Moreover a notation primarily based on a selection of UML notational elements is part of ALEM-N. In the view concept a fundamental distinction is made between a static and dynamic (sub-) model. The static model describes the structure, the dynamic model the behaviour of the modelled system, following the basic distinction in UML (Unified Modelling Language, OMG (2006)) that is also appropriate here.

The Structure View showing the relevant logistic objects is the starting point. The basic elements for this view are UML class diagrams. Besides objects and classes the structure view can show relationships between them, for instance in the form of associations or inheritance relationships.

The Knowledge View describes the knowledge, which has to be present in the logistic objects to allow a decentralized decision making. This view focuses on composition and static distribution of the knowledge while not addressing temporal aspects. For this purpose UML class diagrams and Knowledge Maps are sufficient.

The Ability View depicts the abilities of the individual logistic objects. Processes of a logistic system need certain abilities, which have to be provided by the logistic objects. These abilities are supposed to be seen as abstractions of problem types and their solving capabilities occurring in reality.

The Process View depicts the logic-temporal sequence of activities and states of the logistic objects. Here the objects' decision processes can be modelled. The notation elements used for this are activity diagrams as well as state diagrams.

The Communication View presents the contents and temporal sequence of information exchange between logistic objects. To display the communication UML sequence diagrams showing the interacting partners, the messages and their temporal progression as well as class diagrams to display communication contents are supposed to be used.

The connections between the views are based on the ones defined in the UML meta model (OMG 2006) but are also extended to better guarantee model consistency while additionally restricting freedom during model construction.

3.2. Procedure Model

The procedure model ALEM-P is a guideline for modelling autonomous logistic processes, which contributes on one hand to the assurance of model quality and on the other hand to the reduction of the effort during model construction. It is a specific procedure model, which recommends operational activities using the notational elements and concepts described before. Thereby a system modeller with deepened knowledge about logistics planning and control is enabled to construct a semi-formal system specification to support analysis, design and improvement of systems based on autonomous control.

The procedure model defines steps to pass during model construction, therein activities to perform and results to get out of every step. Furthermore methods and instruments are recommended to support the work. Among these are firstly the presented view concept and diagrams, secondly modelling conventions in terms of construction and consistency rules and thirdly existing techniques suitable for the individual steps. Additionally there are indicators given for necessary iterations that may be initiated in a step, which cause a reengineering cycle by referring to a former step. Basically the procedure is inspired by the top down principle because the system and the enclosed processes are examined on a rather abstract level before they are detailed and concretised. However the focus on selected autonomous logistic objects and their reciprocal coordination with each other as well as the other system elements involves a high importance of the bottom up principle. Thus the procedure is a combination of top down and bottom up approach.

The first step in the specification procedure for autonomous logistic systems broaches the issue of objectives in the system. Starting point are the global system objectives. For a production system the classic goals of production logistics shall be used, from which more concrete local goals can be derived. The documentation of objectives is done in the knowledge view by using class diagrams.

The second step of the specification procedure is the design of the system structure and therewith the collection and documentation of the system elements and their static relations. Central to this step are the autonomous logistic objects - the modeller has to plan which system elements shall have autonomous abilities and which ones not. This aspect will afterwards be further elaborated in the next step. The modelling of the structure is done in the structure view using class diagrams.

The third step of the modelling procedure aims at a structuring of abilities and their mapping to the different logistic objects. Abilities are interpreted as abstract collections of operations that enable an autonomous logistic object to perform certain planning and control tasks. Abilities are modelled in the ability view using class diagrams and especially the concept of interfaces.

The fourth step concentrates on the modelling of the processes running in the system, especially the necessary control processes. The process design is separated in two sub-steps. First routine processes assuming a progression without disturbances are modelled and afterwards these are systematically complemented by processes for handling disturbances and unplanned events. The modelling is done in the process view using activity diagrams and state machines.

The fifth step of the modelling procedure focuses on the decisions. To support identification and adequate description of decisions the structuring of a decision model from decision theory is adapted here (Bussmann 2004, Laux 2005). Thus a control decision can be characterised by a decision maker, an objective and a decision rule representing the objective, a trigger as well as a decision space. The modelling is done in the process view, in particular using activity diagrams.

In the sixth step the focus lies on the knowledge needed for decision making. For that purpose every decision has to be analysed what knowledge is needed. The explicit consideration in the process model is carried out in activity diagrams using object nodes. After examining what knowledge is needed, it has to be specified where it comes from by allocating the knowledge objects. The important point is not the location of knowledge usage, what has been relevant during examination of the decision processes. In contrast it has to be specified where the knowledge objects are available in constantly updated form and thus where demanding autonomous logistic objects can access it. For modelling the allocation of information objects knowledge maps are used.

The communication is modelled in step seven. Thereby two main aspects have to be distinguished. On one hand there are the communication processes and on the other hand the exchanged messages. The modelling of communication is done using sequence and class diagrams. An example for a sequence diagram is shown in figure 1. This communication protocol between machines and commodities is used for the allocation of commodities to machines. The figure shows the sequence of message exchange where the commodity requests a machining operation answered by the machine with a quote containing the possible completion date. The commodity selects one machine and sends an notification after arrival. After the machine has checked its own ability of processing the arrived commodity it sends an arrival notification or a refusal of acceptance.

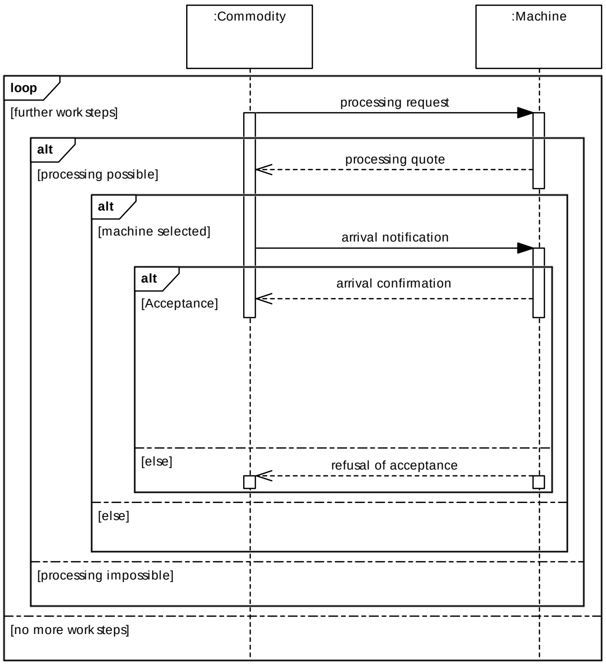

Figure 1: Sequence Diagram for the Allocation of Commodities to Machines

In the eighth step the concrete scenario data is collected. For the classes defined during the previous steps all instances have to be documented to form the basis for the succeeding simulation phase and in the end for the operability of the system. The data is entered in simple lists or matrices. Moreover in ALEM-T it is possible to show at least the resources of the logistic system in a layout diagram and to enter the data there.

4. CASE STUDY

We applied the modelling method just presented to an industrial case study from a manufacturer of automobile supplies. Following the eight step-procedure to derive a model we developed a to-be concept for the control of this production logistic system based on the principle of autonomous control.

In order to assess the operability of the model and it’s logistic performance we derived a multi-agent-based simulation model to simulate the scenario, both without and also with the influence of disturbances. To implement the simulation model we used the multi-agent simulation environment SeSAm (Shell for Simulated Agent Systems) (Kluegl 2006). The most important reason for choosing SeSAm was that it provides visual agent modelling using an UML-based notation. Thus further usage of conceptual models constructed during the specification phase is eased and ensured. Unfortunately the performance of the created simulation model turned out not to be sufficient to run the experiments we intended. Especially determining proper buffer/stock sizes (see the text below) required numerous simulation runs. To be able to perform these simulations we implemented the model and production control methods in our own, Java-based discrete event simulation kernel.

The simulation model consists of 34 machines, 10 different products are manufactured. The material flows within the system are re-entrant, i.e. products have to visit certain machines more than once. Furthermore on some machines sequence-dependent setup times occur. There are further restrictions on machine capabilities: there are alternative machines for most production steps, but there are also restrictions regarding the product variants each machine can produce.

The company our case study is based on produces valve spring holders. A valve spring holder couples the valve spring with the valve in an automobile engine and serves as support for the valve spring. The production of valve springs includes several production steps with different production technologies (see figure 2). For our simulations we selected 10 out of 100 different variants of valve spring holders on the basis of an ABC analysis that cover around 75% of the overall production volume. The differences of the product variants in size and/or material result in different processing times on some production steps, especially in off-pressing, annealing and pressing. So the overall processing time varies between 945 minutes and 1045 minutes per standard lot. Moreover setup times have to be taken in account on off-pressing and pressing machines as well as on packing machines when different product variants are processed successively.

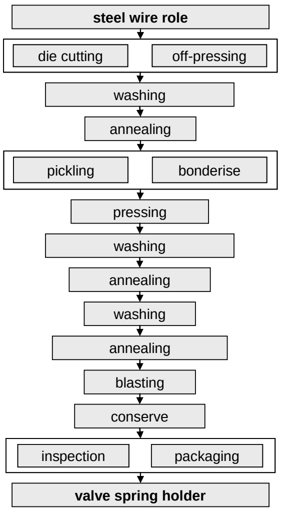

Figure 2: Case Study Production Process

The allocation of commodities to the machines is done as follows. Before each production step a commodity requests processing quotes from all potential machines for the next work step. Machines check the requests and in case of suitability calculate possible processing dates depending on their state and schedule and send back processing quotes containing an estimated time when the requesting commodity can be finished. A commodity selects the machine with the earliest possible date. In the following we refer to this method of production control as *Method A.*

We compare this method with a variant of the well-known KANBAN method (Ohno 1988), in the following referred to as *Method B.* KANBAN was chosen because it is also a decentralized and widely known method for production control. It is furthermore quite simple to implement – its use is even possible without IT-support.

We aim at a zero lead time for customer orders, i.e. arriving customer orders should be processed with a very low flow time close to zero. To minimise costs these low flow times have to be achieved with a work in progress (WIP)-level as low as possible. To achieve this, we have to establish a stock of proper size for finished goods for Method A and properly dimensioned buffers for Method B. The sizes finally used for the results in section 6 are given in table 2, “Stock Size” is the stock size of finished goods used for Method A, the KANBAN buffer sizes for Method B are shown in column 4. The column for KANBAN only contains the sum of the buffer sizes over all 12 production steps of the respective product. Stocks and buffers were dimensioned to achieve a very low mean flow time at a long-term expected utilization of 85% (no downtimes).

To asses the performance of our modelled and implemented method A, we considered two factors, reflecting external or internal sources of disturbances: different demand levels and machine downtimes. Different demand levels are simulated by different bottleneck utilizations of 80, 85, 90 and 95%. Machine breakdowns are considered with three levels: no downtimes, mean time between failures 2 days, mean time between failures 1 day. For the latter two settings we use an exponential distribution with the respective mean.

5. SIMULATION METHODOLOGY

As already stated we simulate 34 machines organized in 10 machine groups. 2 machine groups have to be visited twice so there are 12 operations to finish each lot. We assume a fixed product mix (table 2) but actual demand can vary between 80 and 95% bottleneck utilization based on the product mix shown in table 2. Inter-arrival times are exponentially distributed.

Table 2: Product Mix and Buffer/Stock Size used

Product	Frequency	Stock Size	Kanban Buffer Sizes (Sum)
1	18,1%	14	18
2	16,8%	14	18
3	13,1%	11	13
4	12,3%	10	11
5	11,2%	11	13
6	8,7%	10	11
7	5,4%	8	8
8	5,2%	7	8
9	4,9%	7	7
10	4,3%	6	7
Sum	100,0%	98	114

We are interested in steady state performance of the production system and our main concern is the mean flow time of a lot, i.e. the time required from the arrival of a customer order until the time the order can be fulfilled, as well as the WIP-level in the factory, measured in number of lots.

We therefore simulate a time span of 5 years of continuous production, but to avoid bias only the last 4 years are used to produce the results. We perform at least 5 independent replications of our simulation experiments and if required further replications until we get a confidence interval of at most ± 1% mean flow time at a confidence level of 95% (Law 2007) or alternatively ± 1 minute, whichever is larger.

6. SIMULATION RESULTS

Using the experimental setup just described we achieved results concerning WIP levels and flow times as shown in figure 3 and table 3 respectively.

As can be seen in figure 4 WIP levels for Method A are constant and independent of utilization or breakdowns. It is always equal to the sum of the stock sizes (see table 2), because as soon as a finished good is removed from stock, a new lot is started to refill the stock – the total number of lots on the shop floor and in the stock remains the same. For method B, KANBAN, however WIP decreases with an increasing utilization and increasing frequency of machine breakdowns due to an increasing number of buffer spaces awaiting to be refilled.

Concerning flow times we achieved the results of table 3. For the factor combination used to determine buffer/stock sizes (85% utilization, no breakdowns) both method A and method B show very low results (nevertheless method A needs a lower WIP-level to achieve this. Method A, based on autonomous control, however is clearly more robust with respect to an increasing utilization and more frequently occurring machine breakdowns.

Another criterion is the effort required to find proper parameters for the production control methods. Here again method A outperforms method B as it only requires 10 parameters (the stock size for each finished good) to be set, whereas for KANBAN the size of 120 KANBAN buffers (10x12, one potential buffer for each product and production step) has to be set appropriately.

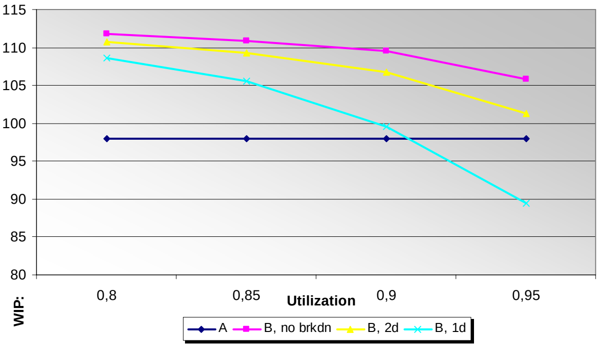

Figure 3: WIP Levels Achieved

Table 3: Flow Times (in Minutes) for Different Settings of Breakdowns and Utilizations

		Method A			Method B		
		none	**2d**	**1d**	**none**	**2d**	**1d**
Utilization	**80%**	0	0	2	1	3	12
	85%	1	1	10	3	8	58
	90%	2	7	85	9	37	393
	95%	33	87	1709	95	246	8432

7. SUMMARY AND OUTLOOK

This paper addressed the topic of modelling autonomous logistic processes. Therefore, after a short definition of autonomous control in the context of logistics, we discussed requirements to a modelling method in this context and evaluated existing methods for this purpose. Afterwards we briefly sketched our modelling method ALEM, specially targeting the design of autonomous logistic processes.

We applied our method to an industrial scenario (section 4) and used this model to derive a simulation model to asses operability and logistic performance of the modelled system. In section 6 we compared the system's performance with the well-known KANBAN-method for production control.

Further work will concentrate on further integrating the modelling and simulation phases and allow a semi-automatic transformation of the conceptual ALEM-N-model into a simulation model, a step currently performed manually. Furthermore we plan to further improve our modelling method by applying it to additional industrial scenarios.

ACKNOWLEDGMENTS

This research is funded by the German Research Foundation (DFG) as part of the Collaborative Research Centre 637 Autonomous Cooperating Logistic Processes - A Paradigm Shift and its Limitations (SFB 637).

REFERENCES

Bauer, B., Mueller, J.P., Odell, J., 2001. Agent UML: A formalism for specifying multiagent software systems. *International Journal of Software Engineering and Knowledge Engineering*. 11 (3): 207–230.

Boese, F., Windt, K., 2007. Business Process Modelling of Autonomously Controlled Production Systems. In: M. Huelsmann, K. Windt, eds. *Understanding Autonomous Cooperation & Control in Logistics - The Impact on Management, Information, Communication and Material Flow*. Berlin: Springer, 73–83.

Bresciani, P, Giorgini, P., Giunchiglia, F., Mylopoulos, J., Perini, A., 2004. Tropos: An Agent-Oriented Software Development Methodology. *International Journal of Autonomous Agents and Multi Agent Systems*, 8 (3): 203–236.

Bussmann, S., Jennings, N.R., Wooldridge, M., 2004. *Multiagent Systems for Manufacturing Control. A Design Methodology*. Berlin: Springer.

Cernuzzi, L, Juan, T., Sterling, L., Zambonelli, F., 2004. The Gaia Methodology. In: F. Bergenti, ed. *Methodologies and Software Engineering for Agent Systems. The Agent-Oriented Software Engineering Handbook*. Boston: Kluwer Academic Publishers, 69–88.

Dangelmaier, W., 2001. *Fertigungsplanung - Planung von Aufbau und Ablauf der Fertigung. Grundlagen, Algorithmen und Beispiele*. Berlin: Springer.

DeLoach, S.A., 2004. The MaSE Methodology. In: F. Bergenti, ed. *Methodologies and Software Engineering for Agent Systems. The Agent-Oriented Software Engineering Handbook*. Boston: Kluwer Academic Publishers, 107–126.

Ferstl, O.K., Sinz, E.J., 2006. *Grundlagen der Wirtschaftsinformatik*. Muenchen: Oldenbourg.

Giret, A., Botti, V., Valero, S., 2005. MAS Methodology for HMS. In: V. Marik, ed. *Holonic and Multi-Agent Systems for Manufacturing. Proceedings of 2nd International Conference on Industrial Applications of Holonic and Multi-Agent Systems, HoloMAS 2005*, 39–49. August 22-24 2005, Copenhagen, Denmark.

Giret, A., Botti, V., 2006. From system requirements to holonic manufacturing system analysis. *International Journal of Production Research*. 44(18-19): 3917–3928.

Haberfellner, R., 2002. *Systems Engineering*. Zuerich, Switzerland: Industrielle Organisation.

Hall, K.H., Staron, R.J., Vrba, P., 2005. Experience with Holonic and Agent-Based Control Systems and Their Adoption by Industry. In: V. Marik, ed. *Holonic and Multi-Agent Systems for Manufacturing. Proceedings of Second International Conference on Industrial Applications of Holonic and Multi-Agent Systems, HoloMAS 2005*, 1–10. August 22-24 2005, Copenhagen, Denmark.

Huelsmann, M., Windt, K., 2007. *Understanding Autonomous Cooperation & Control in Logistics – The Impact on Management, Information, Communication and Material Flow*. Berlin, Germany: Springer.

Iglesias, C.A., Garijo, M., Gonzalez, J.C., Velasco, J.R., 1998. Analysis and Design of Multiagent Systems Using MASCommonKADS. In: M.P. Singh, ed. *Proceedings of the 4th International Workshop on Agent Theories, Architectures, and Languages (ATAL'97)*, 313–327. July 24-26, 1997 Rhode Island, USA.

Kluegl, F., Herrler, R., Fehler, M., 2006. SeSAm: Implementation of Agent-Based Simulation Using Visual Programming. In: *Proceedings of the AAMAS 2006*, 1439-1440. May 8-12 2006 Hakodate, Japan.

Laux, H., 2005. *Entscheidungstheorie*. Berlin: Springer.

Law, A., 2007. *Simulation Modeling and Analysis*, 4th edition. Boston et al.: McGraw-Hill.

McFarlance, D.C., Bussmann, S., 2003. Holonic manufacturing control: rationales, developments and open issues. In: S.M. Deen, ed. *Agent-Based Manufacturing - Advances in the Holonic Approach*, Berlin, Germany: Springer, 303–326.

Mertins, K, Jaekel, F.-W., 2006. MO2GO: User Oriented Enterprise Models for Organisational and IT Solutions. In: P. Bernus, ed., *Handbook on Architectures of Information Systems*. Berlin, Germany: Springer, 649–663.

Monostori, L., Vancza, K., Kumara S.R.T., 2006. Agent-Based Systems for Manufacturing. *CIRP Annals*, 55(2): 697–720.

Odell, J.J., Parunak, H.V.D., Bauer, B., 2001. Representing agent interaction protocols in UML. In: P. Ciancarini, M. Wooldridge, eds. *Agent Oriented Software Engineering*. Berlin: Springer, 121-140.

Oestereich, B., Weiss, C., Schröder, C., Weilkiens, T., Lenhard, A., 2003. *Objektorientierte Geschäftsprozessmodellierung mit der UML*. Heidelberg: Dpunkt.

OMG, 2006. *Unified Modeling Language Specification, Version 2.0*. Available from: www.uml.org (accessed 01 December 2006).

Parunak, H. Van D., Sauter, J., Clark, S., 1998. Toward the Specification and Design of Industrial Synthetic Ecosystems. In: M.P. Singh, ed. *Intelligent Agents IV: Agent Theories, Architectures, and Languages*. Berlin: Springer, 45–59.

Scheer, A.-W., 2000. *ARIS - Business Process Modeling*. Berlin: Springer.

Scholz-Reiter, B., Kolditz, J., Hildebrandt, T., 2007. Engineering autonomously controlled logistic systems. *International Journal of Production Research:* 1-20.

Schreiber, G., Akkermans, H., Anjwerden, A., de Hoog, R., Shadbolt, N., van de Velde, W., Wielinga, B., 2001. *Knowledge Engineering and Management - The CommonKADS Methodology*. Cambridge, USA: MIT Press.

Specker, A., 2005. *Modellierung von Informationssystemen. Ein methodischer Leitfaden zur Projektabwicklung.* Zuerich: vdf Hochschulverlag.

Ohno, T., 1988. *Toyota production system: beyond large-scale production*, Portland, USA: Productivity Press.

Valckenaers, P., Heikkilä, T., Baumgaertel, H., McFarlane, D., Courtois, J.-P., 1999. Towards a novel manufacturing control principle. In: H. Van Brussel, ed. *Proceedings of the Second International Workshop on Intelligent Manufacturing Systems*, 871–875. September 22-24, 1999 Leuven, Belgium.

Vernadat, F., 2006. The CIMOSA Languages. In: P. Bernus, ed. *Handbook on Architectures of Information Systems*. Berlin: Springer, 251–272.

Windt, K., 2006. Selbststeuerung intelligenter Objekte in der Logistik. In: M. Vec, ed. *Selbstorganisation - Ein Denksystem fuer Natur und Gesellschaft*, Cologne, Germany: Boehlau, 271–314.

Wooldridge, M. and Jennings, N.R., 1995. Intelligent agents: theory and practice. *The Knowledge Engineering Review* 2 (10): 115–152.