

Analysis and Design of Autonomous Logistic Processes

B. Scholz-Reiter¹ (2), J. Kolditz¹, T. Hildebrandt¹

¹ Department of Planning and Control of Production Systems
University of Bremen, Bremen, Germany

Abstract

Control of dynamics and complexity of logistic systems will continue to gain in importance in the future. One possibility to cope with this challenge is the concept of autonomous logistic processes that counts on using sophisticated smart labels. This article addresses the system development process and presents a modelling methodology for analysis and design of autonomous logistic processes. It will give an overview of the created modelling and view concepts and details selected aspects of them.

Keywords:

autonomous logistic processes, modelling, modelling concept, system development process

1 INTRODUCTION

Today enterprises are exposed to an increasingly dynamic environment. Last but not least increasing competition caused by globalisation more and more requires gaining competitive advantages by improved process control, within and beyond the borders of producing enterprises. One possibility to face increasing dynamics is autonomous control of logistic processes. This shall allow more robust processes in spite of growing environmental as well as internal complexity. This paper presents the idea of autonomous logistic processes and focuses on a concept for modelling such processes. First section 2 gives a short overview of the concept of autonomous logistic processes. Section 3 presents the overall system development cycle. Main section 4 discusses process modelling under the paradigm of autonomy and introduces important aspects of our modelling method. The paper is concluded by a short summary and an outlook of future work.

2 AUTONOMOUS CONTROL OF LOGISTIC PROCESSES

Autonomous control in the context of SFB 637, the research project this work is based on, means processes of decentralized decision making in heterarchical structures. It requires the ability and possibility of interacting system elements to autonomously make goal-oriented decisions. The use of autonomous control aims at achieving a higher robustness of systems and simplified processes achieved by distributed handling of dynamics and complexity due to greater flexibility and autonomy of decision making. Focus of the SFB lies in the areas of production and transport logistics, so the system elements, making their decisions autonomously, are the logistic objects themselves [1].

In order to enable logistic objects to be intelligent they have to be provided with smart labels. While today's RFID (radio frequency identification)-chips have very limited capabilities in respect to energy, range, storage capacity and especially information processing [2], near future shall bring highly evolved smart labels that can provide resources alike micro computers to logistic objects. Nowadays RFID is already widely used in industry for identification matters and several visions for future applications exist [3], [4].

With respect to shades of autonomous control, different scenarios are possible, depending on which logistic objects are provided with smart labels and the functionalities they offer. This determines to what extend the logistic objects are able to make decisions. Considering the kind of decision-making by autonomous and therefore potentially intelligent logistic objects, transferring control decisions to goods, machines, storages and conveyors is obvious. Besides scenarios, where only one of the kinds of logistic objects has the ability to autonomously make decisions, arbitrary combinations are possible, depending on whether objects of the respective group are rather autonomously controlled or not.

Different logistic objectives can be assigned to the different groups of objects. For instance the objective of a high utilization can best be assigned to machines, while the objective of low due date deviation can best be assigned to a good. Concrete goal values are only achieved by the interaction of many logistic objects. Often conflicting goals of different objects have to be balanced, e.g. by negotiation. This leads to an increased coordination and communication effort compared to hierarchic forms of finding a decision. The more objects and groups of objects are involved in such a communication and make their decisions autonomously, the more important this point becomes. The number of possible communication relationships roughly grows quadratic in the number of participating objects. With 10 communicating objects there are 45 possible relationships, having 100 objects already leads to 4950. These numbers make clear that communication has to be limited to objects in the immediate spatial and/or logic neighbourhood as otherwise control strategies can only hardly be scaled to problems of a realistic size. All these points have to be considered designing a control strategy and for modelling such a system.

3 DEVELOPMENT OF A LOGISTICS SYSTEM BASED ON AUTONOMOUS COOPERATING PROCESSES

This chapter focuses on the development procedure of a logistic system based on autonomous cooperating processes and the integration of the proposed modelling methodology. The system development is formulated as an iterative process shown in figure 1. The steps 1 and 2 will

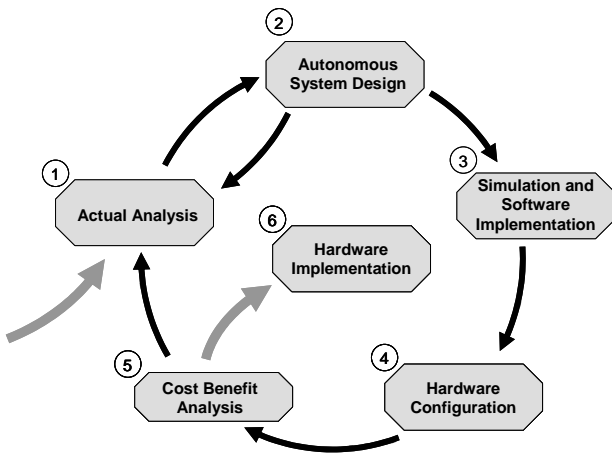


Figure 1: System development process

be supported by the methodology for modelling autonomous logistic processes.

- 1 The starting point is an analysis of the actual state of the system. At the beginning this step is alike a feasibility study. That means very basic issues like an estimation to what extend autonomous cooperating processes are actually suited for the scenario. Associated with that the question has to be answered which objectives are pursued in the particular case by implementing autonomous logistic processes. If the application is not evaluated positively for this scenario the development procedure should be cancelled here. It has to be pointed out that this step forms a mini iteration with the following one. After the system design in step 2 concrete scenario data has to be collected and registered like machines, products or work steps. This step does not affect the degree of autonomy but is the connection between the in principle universal design of the system and its processes and concrete scenario data.
- 2 The next step consists of the design of the system. There a semi-formal specification of the proactive elements in an autonomous system as well as 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 intelligent respectively autonomous entities are. To ensure the operability of the system all elements and processes have to be aligned with each other, making this step the basis of the development procedure.
- 3 During the step of simulation and software engineering the design realised before is tested in a simulation first. Especially operability and impact on logistics performance of the whole system are focused here. A central task is the verification of required system behaviour because this is a necessary precondition for industrial application of emergent systems like autonomous logistic processes. 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 a subsequent iteration loop.
- 4 On the basis of the ideas 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. Conclusions may be drawn from the process model as well as from the simulation. For example from allocation of control processes and data packets to entities of the logistic system necessary memory and computing capacity can be derived. Another example is the prediction of the capacity and

equipment of the communication infrastructure on the basis of the expected communication volume between logistic system entities resulting from the simulation and the physical distribution of the objects to be arranged during hardware configuration. Attention has to be paid to the fact that although several agreements have been done during the steps before, this step strongly impacts implementation costs.

- 5 Now compared to the initial feasibility study a much more significant cost benefit analysis is possible getting more detailed during subsequent iteration loops. Thereafter the original process model can be adjusted in step 2 according to the new conclusions. In case of repeating negative results in this step an application of autonomous logistics processes has to be abandoned for this scenario.
- 6 The final step of the development process is the installation of the system based on autonomous logistic processes. In case of insufficient experience a prototypic setup should be tested before the actual installation.

4 MODELLING AUTONOMOUS CONTROL

4.1 Overview

The modeling method consists of the components illustrated in figure 2. The "Principles", shown in the center of figure 2, define the basic structuring of the method. They consist of a view concept, each emphasizing certain aspects of the system to be modeled, as well as elementary guidelines of modeling. The "Meta Model" specifies the modeling elements usable by the modeler in a view-spanning manner. "Diagrams" defines the graphical notation representing these elements and the contexts where they can occur. It defines different diagrams each focusing on different facets of the system and visualizing them. Some examples of these diagrams are discussed later on in conjunction with discussing the view concept.

On the basis of the defined elements a reference model for autonomous cooperating processes is established. This reference model is available to the modeler as a set of building blocks easing model construction. The business process specialist will also get a modeling tool and the procedure model sketched in steps 1 and 2 of the system development process described in the previous section that is intended to guide the user through analysis and specification of autonomous cooperating logistic processes in the surveyed system.

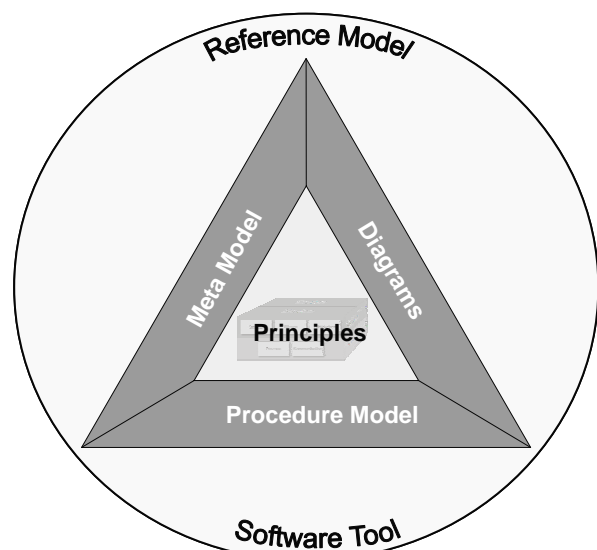


Figure 2: Elements of the proposed modeling method

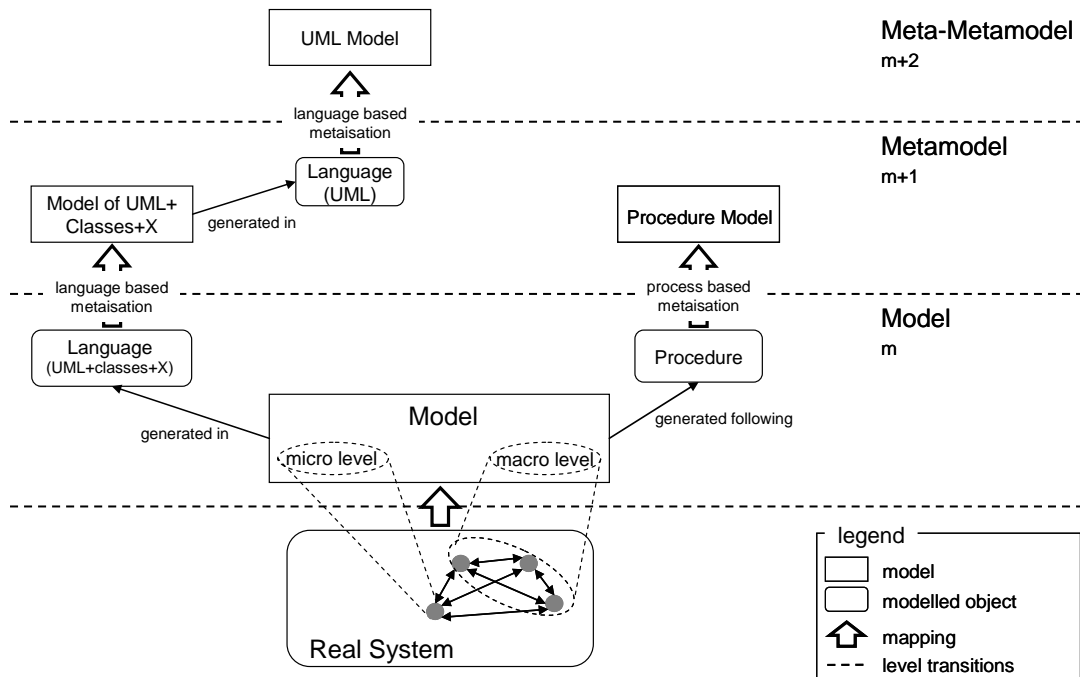


Figure 3: Modelling method in the context of different modelling levels

4.2 Modelling Concept

Before the next part gives further details of the modelling method and the view concept behind it, we will give an overview of the modelling concept on an abstract level, i.e. shown in the context of different modelling levels (see figure 3). The figure shows different modelling levels, from the mapping of the real system at the bottom to the model level as well as from the modelling layers to their respective meta-levels. The distinction between model and meta-model is the same as between the real system level and the model level: the higher level contains explicitly the elements that can be used to model the level below. This means the meta-model-level specifies the elements that can be used to model the system on the model level. Speaking of “elements” this refers only to one aspect of the level transition, the specification of the modelling language. This aspect is called “language-based metaisation” in contrast to “process-based metaisation” which shows the modelling procedure to be used on the level below.

On the lowest layer of figure 3 the (real or thought) system can be found. This is the system to be modeled; the modelling process itself is indicated by the lowest layer transition. Additionally the distinction between a macro- and a micro-level in modelling is indicated. Details regarding this point can be found in the next section. The model on one hand was created in a certain modelling language and on the other hand created following a certain modelling process. Therefore the layer transition from the model to the meta-model-layer distinguishes between language-based and process-based metaisation (for more information on metaisation refer to [5]). Explicit representation of the creation process leads to the depiction of a procedure model for modelling. The procedure model will be represented using natural language and the process of its creation is not of particular interest to us thus nothing is shown in the figure on the meta-meta-model layer regarding the language- or process-based metaisation of the procedure model.

Concerning the branch of language-based metaisation and the transition from model- to meta-model-layer, the modelling language respectively modelling notation is explicated. Our modelling notation is based on version 2.0 of the Unified Modelling Language (UML). In addition to

that the modeller will be supported by pre-defined domain-specific classes and logistic-specific process-parts and process-templates. The UML notation is extended to better show certain aspects of the logistic system, for example by elements taken from software agent modelling. These extensions of the modelling language are indicated in the figure by the “X”.

This (language-based) meta-model again is depicted in a certain way. At this point the distinction between language- and process-based metaisation could be made again, but only the first is of interest here. To represent the modelling notation, as a means of semi-formal modelling, UML will be used. To depict the fact that also this modelling language has to be specified somewhere, the top-layer shows the “model of UML”, being the UML specification (see [6]). Relative to the modelling we aim at, this specification is on the layer of a meta-meta-model, strictly following language-based metaisation.

4.3 Concepts of views

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 [7] which is also reflected in the guideline of systematic design (see subsection Requirements to modelling). Based on the requirements mentioned above a view concept for modelling of autonomous logistic processes is proposed, whose views are depicted in figure 4. A fundamental distinction can be made between a static and dynamic model. The static model describes the structure, the dynamic model the behaviour of the modelled system, according to the basic classification in UML [6] that is also appropriate here.

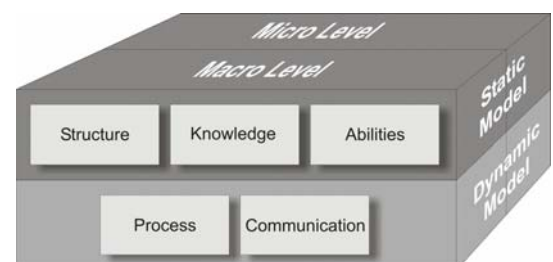


Figure 4: View concept

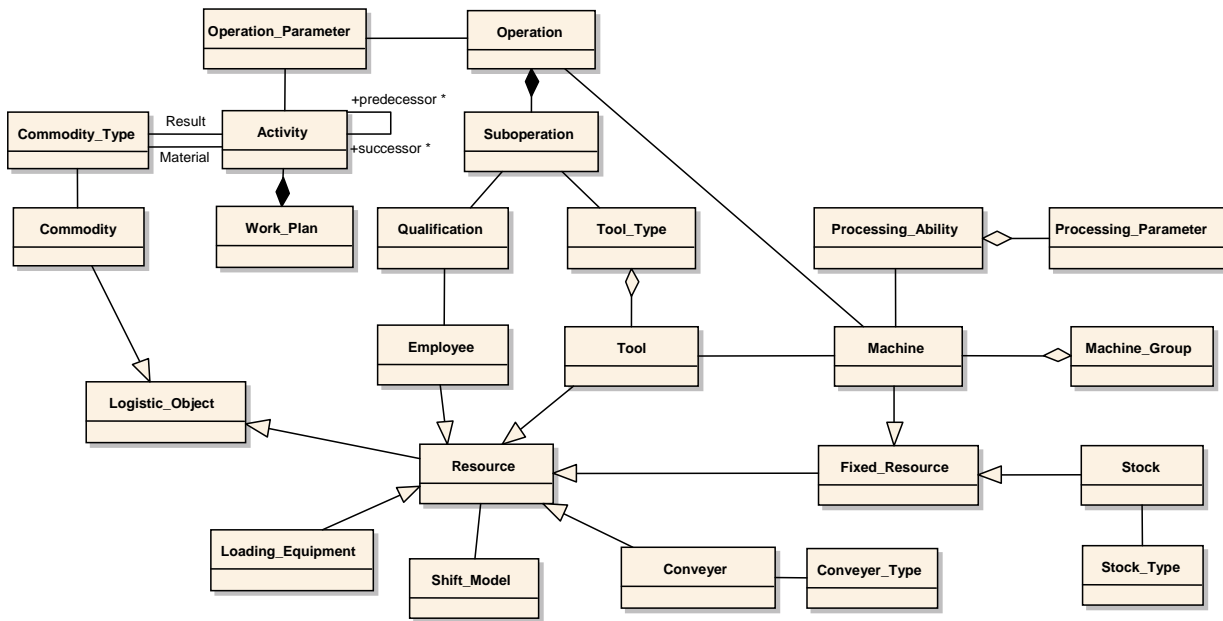


Figure 5: class diagram showing a part of the taxonomy supporting the user and selected relationships between the classes shown

The *structure view* that shows 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 are sufficient, while for the just mentioned temporal aspects, a dedicated knowledge representation language would have to be used [8]. However it is doubtful how far the additional complexity in using it is compensated by the increased expressiveness. This is especially more important with respect to the intended use of the modelling method by a process expert.

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 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 process view plays a central role connecting the views of the static model and depicting the behaviour of logistic objects, so far only viewed statically. The notation elements used for this are activity diagrams as well as state diagrams. These two diagrams are also proposed in business process modelling using the UML [9].

The *communication view* presents the contents and temporal sequence of information exchange between logistic objects. Depicting the communication is especially necessary to depict the interaction of autonomously deciding, otherwise only loosely coupled objects to model their interaction [10]. 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.

In addition to the dynamic and static model just described we distinguish a macro and micro perspective. This distinc-

tion is also used in methods for software agent development [11]. The macro view describes the interaction between the autonomous logistic objects. To some extent, it shows an external view onto the system, its elements and their relations and interactions. On the contrary the micro view describes the actions within and composition of the autonomous logistic objects. For the micro-level especially the process, knowledge and ability view are relevant, while all views proposed are relevant for the macro-level. This means that the micro-macro perspective is orthogonal to the views shown in figure 4. Nevertheless not all views use both perspectives to the same extend.

As an example for the static model and to clarify the described modelling concept figure 5 shows a part of the classes available to the modeller. He can create instances of the existing classes as well as adapt and/or expand the class model. This means that the diagram is a basis that can be adapted for applications of the modelling method if necessary and furthermore be used to model a concrete scenario by creating instances of these classes, e.g. to model actual machines or work plans. The figure shows some relevant classes and the most important relations between them. For clarity reasons there are no multiplicities included in the diagram and most role names as well as attributes of the classes are omitted. To create the collection of domain specific classes [7], [12] and [13] were used as references. The models presented there were used in context of information system development and are now adapted to our requirements of modelling autonomous logistic processes.

As central classes "Logistic Object" and "Resource" (itself being a logistic object as indicated by the inheritance relationship) can be identified. Logistic objects are in principle able to be the autonomous objects of autonomous logistic processes. Kinds of logistic objects are commodity, all types of resources and orders (not shown in the selected classes above). Commodity represents a concrete logistic object in a material flow, e.g. an individual end-product, while commodity type is used when a commodity shall be referred to anonymously. A commodity type might be a type of end product, intermediate product or raw material. Work plans, which are an aggregation of "Activities" specify how a commodity can be manufactured, i.e. which work steps to perform and what the required material(s) are and what the result of such a processing or assembly step is.

This work plan is specified anonymously, i.e. for “Commodity type”s. “Resource” represents a common base class for physical and rather permanent components of a production system, each of them can be associated with a “Shift Model”, which determines resource availability and therefore is an important factor for its capacity. Specialisations of the resource class are machine, tool or stock as well as conveyer, tool, loading equipment and employees, the latter being a software representation or an interface of/to workers on the shop floor.

In order to facilitate a loose coupling of the components of our logistics system there is no static mapping between the activities within a work plan and the machines or other resources to perform them. This is advantageous to achieve a more adaptive behaviour of the system. If new machines are added to the shop floor, they can start processing in a “plug-and-play”-like manner without the necessity to change all existing work plans. Work plans only specify which activity to perform and their parameters, as a simplified example drilling, 5mm wide, 7mm deep. To determine the next machine a commodity asks machines which of them can perform a certain activity. This negotiation process is further specified in the communication and process views. A machine is able to autonomously deduce whether it is able to perform an activity on the basis of its processing abilities stored within it (e.g. able to perform “drilling” in the range of 2-10mm wide, 1-20mm deep). Furthermore it is able to create operations on the basis of activities and processing abilities, which in detail specify which and how long tools and personnel are required to perform such an activity.

As an example for the dynamic model figure 6 shows an exemplary sequence diagram as part of the communication view. The example is rather simplified and concentrates just on commodity-machine communication although availability of conveyers must be considered in a resource selection process. The diagram shows a machine object and a commodity object. The exchanged messages are shown chronologically in vertical direction. The commodity requests a machining process answered by the machine with a quote. After the machine has selected a quote (the selection itself with its criteria and algorithms is modelled in the micro level process view) the chosen machine is booked by the commodity, the others are informed about the quote cancellation. In figure 6 this is modelled by a combined fragment of the type “alternative”.

The presented example also shows some deficits of the UML 2.0 standard with respect to modelling autonomous logistic processes. It is not one commodity communicating with one machine, but one commodity communicating with multiple machines. On the other hand the “maschine selected”-part of the alternative fragment is only executed with one machine. For increased clearness this should be modelled explicitly. One possibility to assure clearness could be an extended notation similar to cardinality which is proposed for software agent modelling with UML using specific extensions [14].

5 SUMMARY

This paper addressed the topic of modelling autonomous logistic processes. Therefore after a short definition of autonomous control in the context of logistics, the overall system development process was sketched. After that the concept of our modelling method was presented, first giving a rough overview, then detailing selected aspects of it such as the view concept.

Further research will detail notations to be used and be concerned with the elaboration of the procedure model. Finally our work will result in the development of a software tool, specifically tailored to support our modelling method comprised of the notation and procedure model as far as

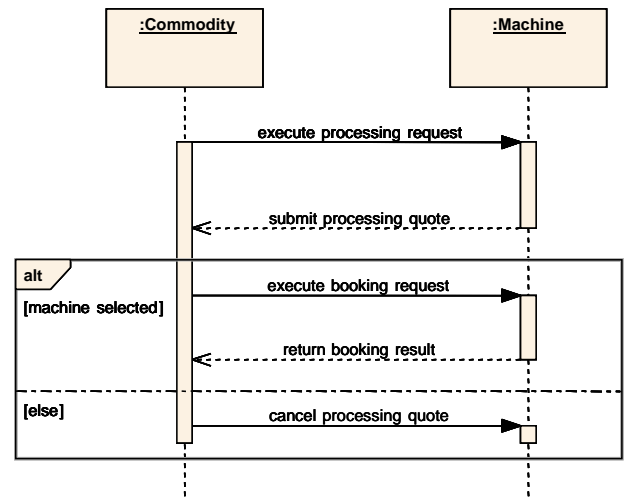


Figure 6: UML sequence diagram machine selection

possible. With the help of this tool a process expert (e.g. a logistics expert with only little background in computer science) will be supported in modelling and designing autonomous logistic processes.

6 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).

7 REFERENCES

- [1] Scholz-Reiter, B., Windt, K., and Freitag, M. “Autonomous Logistic Processes - New Demands and First Approaches” in Proceedings of the 37th CIRP-International Seminar on Manufacturing Systems 2004, Budapest, Hungary, 19.-21.5.2004, pp. 357-362.
- [2] Finkenzeller, K. RFID-Handbook - Fundamentals and Applications in Contactless Smart Cards Identification, Wiley & Sons LTD, 2. edition, Swadlincote, UK, 2003.
- [3] Fleisch, E., and Mattern, F. (eds.) Das Internet der Dinge - Ubiquitous Computing und RFID in der Praxis: Visionen, Technologien, Anwendungen, Handlungsanleitungen. Springer, Berlin, Germany, 2005.
- [4] Heinrich, C.: RFID and Beyond: Growing Your Business through Real World Awareness, Wiley Publishing, Indianapolis, Indiana, 2005
- [5] Strahinger, S. “Probleme und Gefahren im Umgang mit Meta-Begriffen: ein Plädoyer für eine sorgfältige Begriffsbildung“ in Proceedings of the International Knowledge Technology Forum (KnowTechForum) '99. Potsdam, Germany, 1999.
- [6] OMG Unified Modelling Language Specification (version 2.0), Retrieved October 6, 2005, <http://www.uml.org>
- [7] Scheer, A.-W. Business Process Engineering – Reference Models for Industrial Companies, Springer, Berlin et al., 2nd ed., 1994.
- [8] Sowa, J. Knowledge representation: logical, philosophical, and computational foundations, Brooks Cole Publishing Co., Pacific Grove, USA, 2000.
- [9] Oestereich, B. Objektorientierte Geschäftsprozessmodellierung mit der UML, dpunkt Verlag, Heidelberg, Germany, 2003.
- [10] Weiß, G., and Jakob, R. Agentenorientierte Softwareentwicklung, Springer Verlag, Berlin Heidelberg, Germany, 2005.

- [11] Weiß, G. (ed.) Multiagent Systems – A modern approach to distributed artificial intelligence, The MIT Press, Cambridge, USA, 2000.
- [12] Loos, P. Datenstrukturierung in der Fertigung, Oldenbourg, München, 1992.
- [13] Schönsleben, P. Integrales Informationsmanagement: Informationssysteme für Geschäftsprozesse – Management, Modellierung, Lebenszyklus und Technologie, 2. edition, Springer, Berlin, 2001.
- [14] Bauer, B. and Odell, J. "UML 2.0 and Agents: How to Build Agent-based Systems with the new UML Standard" in Journal of Engineering Applications of Artificial Intelligence (18:2), 2005, pp. 141-157