Proceedings of DET2007
4th International Conference on Digital Enterprise Technology
Bath, United Kingdom
19-21 September 2007

THE AUTONOMOUS LOGISTICS ENGINEERING METHODOLOGY (ALEM)

Bernd Scholz-Reiter
Department of Planning and Control of Production Systems, University of Bremen, Germany
bsr@uni-bremen.de

Jan Kolditz
Department of Planning and Control of Production Systems, University of Bremen, Germany
kol@uni-bremen.de

Torsten Hildebrandt
Department of Planning and Control of Production Systems, University of Bremen, Germany
hil@uni-bremen.de

ABSTRACT

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 an enterprise. Autonomous control of logistic processes is proposed as a means to better face dynamics and complexity. Autonomous control means the ability of logistic objects to process information, to render and to execute decisions on their own. To engineer logistics systems based on autonomous control, dedicated methodologies are needed. This paper proposes a methodology for system specification that consists of a notational part, a procedure model and a software tool, covering a substantial part of the overall system engineering process. Supported by this methodology a logistics process expert shall be enabled to specify an autonomous logistic system adequately.

KEYWORDS

autonomous control, process modelling, production control, systems engineering

1. INTRODUCTION

Improved process control, within and beyond the borders of producing enterprises becomes more important because of an increasing competition caused e.g. by globalization and the exposure of enterprises to an increasingly dynamic environment. 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. In the context of the collaborative research centre named SFB 637, the research project this work is based on, autonomous control means the ability of logistic objects to process information, to render and to execute decisions on their own (Hülsmann and Windt 2007). The objective of autonomous control is the achievement of increased robustness and positive emergence of the total system due to distributedly and flexibly coping with dynamics and complexity. As the focus of the research project lies in the areas of production and transport logistics, the system elements making their decision autonomously are the logistic objects like commodities, machines, storages and conveyors themselves (Scholz-Reiter et al. 2004). In order to enable logistic objects to be "intelligent" they have to be provided with smart labels. Nowadays RFID (radio frequency identification)-labels are already widely used in industry for identification matters but have very limited capabilities with respect to, range, storage and energy capacity and especially information processing abilities (Finkenzeller 2003). Near future shall bring highly evolved smart labels (Fleisch 2005, Heinrich 2005) that can provide resources alike micro computers to logistic objects, allowing the "pure vision" of autonomous logistic processes to be realized.

To develop such a system requires a special engineering methodology to properly design all necessary aspects of the system, such as: how does the scenario look like, what logistic objects are there, how much "intelligence" do they have, what situations do they have to cope with? This information is the basis to specify a suitable control strategy, the necessary processes and decisions as well as the communication or other coordination mechanisms between the autonomous logistic objects.

The engineering of an autonomous logistics system can be described on the basis of the general Systems Engineering procedure model (Haberfellner 2002), as shown in figure 1. In the following the cycle shown on the right in figure 1 as the methodical core of the engineering process is sketched.

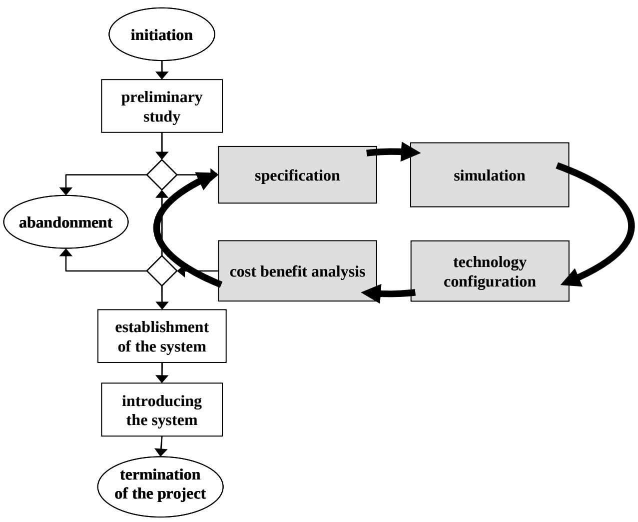

Figure 1 – Engineering autonomous logistics systems (in extension to Haberfellner, 2002)

The first step of the cycle consists of the *specification* of the system. In this step a model of the system is created in the form of a semi-formal specification of the autonomous logistic objects in the 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 are "intelligent" respectively autonomous entities. To ensure the operability of the system all elements and processes have to be aligned with each other, making this the basic step.

During the step of *simulation* the design created 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 logistics systems. 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 during establishment phase later on.

On the basis of the insights gained in the two preceding steps an estimation of needed hardware equipment for the autonomous system (for example what kind of communication infrastructure is needed to allow the necessary communication between the elements) 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.

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

The focus of this paper is a modeling methodology supporting the specification step as the basis of the cycle. The paper is structured as follows: The next section starts with requirements to a modelling methodology for specifying the autonomous logistic system. Additionally existing approaches are evaluated concerning these requirements. The main section 3 introduces the modelling methodology and details the procedure model as an important aspect of it. After a discussion to what extend the proposed modelling methodology fulfils the requirements derived in section 2, the paper is concluded by a short summary and an outlook of future work

2. REQUIREMENTS AND STATE OF THE ART

Formulating requirements for a methodology for modelling autonomous logistic processes has to start on the basis that the modelling methodology meets the definition of a modelling methodology in general. A modelling methodology is a systematic approach which defines the essential modelling referred tasks within one or several phases of a development process. It includes a basic structuring to better handle modelling complexity, a notation the model is constructed with and a procedure model that serves the goal-oriented modelling process.

The main requirement regarding content is the support for construction of models that represent the constitutive attributes of autonomous logistic processes. These central attributes can be derived from the definition of autonomy in logistics from above.

Fundamentally the paradigm of autonomy in logistics is characterised by high importance of the single logistic objects, what calls for a specification focused on them and therefore implicates a bottom-up instead of a top-down approach.

The criterion of information processing determines the possibility to on one hand specify the information processes and on the other hand to allocate them to the performing logistic objects.

The attribute of an autonomous logistic object to render decisions itself causes the necessity of modelling the allocation of decisions as well as the decisions themselves including the corresponding aspects, like knowledge the decision is based on, or the objective(s) pursued with a decision.

The criterion of decision execution results in a need for adequate synchronisation of material and information flow to on one hand assure realisation of decisions of logistic objects in the material flow and on the other hand to allow them monitoring the progress by observing the environment.

In a heterarchically structured system of autonomous elements, like an autonomous logistics system, intensive interaction of the system elements is required to coordinate their actions, what results in high importance of communication (Malone, 1994), (Weiss, 2005). An additional requirement for the methodology is therefore the possibility to model the communication of the system elements.

The task of constructing a model of autonomous logistic processes shall primarily be assigned to an expert for planning and control of logistic processes. This results in a qualification profile that is the orientation for designing the methodology because additional skills for using it have to be minimised. Alongside to the user orientation more requirements concerning the use of the models can be derived from the sketched engineering process for autonomous logistic systems. Thus the model is the basis for software engineering and simulation respectively. This transfer to the software implementation has to be incorporated in the design of the modelling methodology.

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. However in spite of the numerous existing methodologies for agent oriented software engineering, see e.g. (Weiss and Jakob 2005), 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). 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 (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 decision making being a basis 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.

In context of software engineering, methodologies for business process modelling are intended to support the development of centralised information systems (Scheer, 2001). Because of this purpose dedicated concepts for specifying decentralised approaches, instruments for detailed illustration of local information processing and particularly techniques for explicit modelling of communication processes and protocols are missing.

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 modeled 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 (Oestereich, 2005).

3. MODELLING AUTONOMOUS CONTROL

This section introduces a modeling methodology supporting a semi-formal specification of the autonomous logistic system. The modeling methodology as part of the Autonomous Logistics Engineering Methodology (ALEM), consist of the components ALEM-N (ALEM-Notation), ALEM-P (ALEM-Procedure) and ALEM-T (ALEM-Tool). 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 logistic expert 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. Furthermore a reference model is also part of ALEM and offered by ALEM-T to ease the construction of a new model by reusing existing work.

3.1. ALEM-N: THE CONCEPT OF VIEWS AND THE NOTATIONAL ELEMENTS

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, 2001). Based on the requirements mentioned above a view concept for modelling of autonomous logistic processes is proposed, distinguishing five different views as shown in figure 2. A fundamental distinction can be 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.

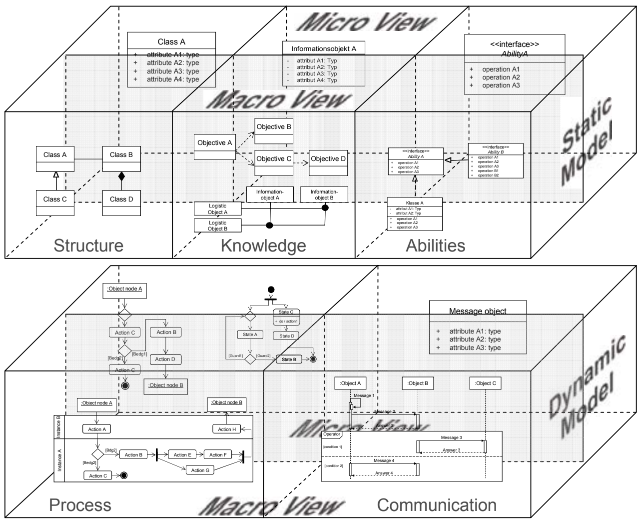

Figure 2 – View Concept

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 (Allweyer, 1998) are sufficient, while for the just mentioned temporal aspects, a dedicated knowledge representation language would have to be used (see e.g. Sowa (2000)). However it is doubtful how far the additional complexity in using it is compatible with 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 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 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 (Oestereich et al, 2003).

The *Communication View* presents the contents and temporal sequence of information exchange between logistic objects. Depicting the communication is especially necessary to describe the interaction of autonomously deciding, otherwise only loosely coupled objects to model their interaction (Weiss, 2000). 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 orthogonal to the views. This distinction is also used in methods for software agent development (Weiss, 2000). The macro view describes the interaction between the autonomous logistic objects. To some extend, 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.

3.2. ALEM-P: THE PROCEDURE MODEL

The procedure model 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 steps of ALEM-P are shown in figure 3 and described subsequently.

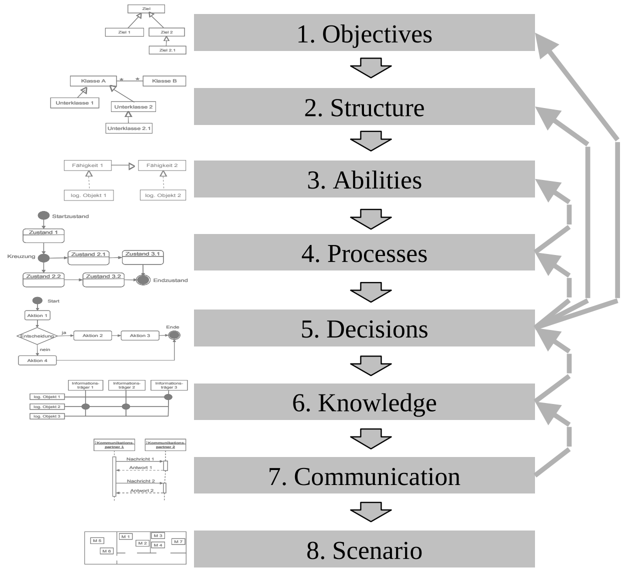

Figure 3 –Procedure Model

3.2.1. Objectives

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 that have to be clarified by the modeller if needed in cooperation with a person in charge of strategic topics. 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 documented and structured objectives as a result of this step are revisited, if necessary detailed and allocated to autonomous logistic objects in subsequent steps of the procedure, especially during examination of the decisions.

3.2.2. Structure

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. So the system construction is not built purely top down, but starts with selected logistic objects that are intended to have autonomous abilities and which are connected with other system elements, thereby gradually building the complete system structure. The modelling of the structure is done in the structure view using class diagrams.

3.2.3. Abilities

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. An ability and therewith the realising operations may be structured themselves since an ability can consist of several sub-abilities. Abilities are modelled in the ability view using class diagrams and especially the concept of interfaces. In early phases of model construction this has to be a rather rough concept of the abilities and their allocation to logistic objects. With ongoing iterations a more and more complete collection of necessary operations in the system is intended, therefore a continuous update and completion of the abilities structuring and mapping is essential. Thereby an easier identification of function accumulation is possible, which may require an adjustment and reallocation of abilities, if restrictions, for example caused by limited computing capacity for specific autonomous logistic objects, are violated. Thus during this step on one hand the agenda for the following process design is set and on the other hand there may and have to be adjustments in the mapping and structuring according to subsequent findings.

3.2.4. Processes

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. As the starting point the processes of the logistic objects running in the physical system are used as an orientation. In a production system for example the way of a commodity through shop floor from inbound storage to outbound storage, the processes that have to be performed by a machine in the course of time or the activities and states of a conveyor from loading commodities to planning its route have to be examined. Thereby the processes that are in principle performed by all instances of a class are essential, not the actions performed by only few objects. Thus the possible states of the logistic objects have to be described on different levels of abstraction and on that basis the operations to be performed in these states. A detailed specification of the decisions that have to be made is not intended yet, but their integration in the surrounding processes. According to this, decisions that are in principle characteristic for control processes (Dean and Wellman, 1991), at this stage of the procedure are modelled as rather abstract activities, named with adequate terms like *choose* or *decide*. An additional simplification is the presumption of unrestricted information. This allows to initially disregard the necessary information acquisition and thus to put back the determination of information sources. Outputs of this sub step are control processes necessary for a system running without disturbances, assuming availability of all information needed for decisions. During the second sub-step the presumption of an ideal system including the disregard of disturbances is replaced by the consideration of uncertainties existing in a logistics system. For the design of an autonomous logistic system therefore the processes already modelled earlier have to be complemented by control processes for handling disturbances. To ensure a systematic integration of relevant processes existing classifications of causes, disturbances and effects as well as adequate tasks of disturbance management like in Patig (2001) are used as an orientation. Especially the disturbances have to be considered that shall later be handled automatically in the system, allowing their handling by the autonomous logistic objects, independent of the date of disturbance occurrence. However also disturbances have to be taken into account which require human intervention, for example to be able to provide adequate human-machine interfaces.

3.2.5. Decisions

This 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). The modelling is done in the process view, in particular using activity diagrams, and in the knowledge view using class diagrams as well as knowledge maps. For identification of decisions all autonomous logistic objects and the process models constructed in step four have to be examined. Basis of identification and characterisation is the structure of a decision. 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 decision space is determined by possible activities and the consequences connected to the activities. Beyond identification by their structure decisions can normally be isolated in the constructed model by intentionally as well as unintentionally naming of relevant activities like “choose…” or “decide…”.

Besides the decisions already considered in the processes modelled before more decisions are found on the basis of decision dependencies. In that case a corresponding extension of the process models is necessary. After identification of the different decisions they have to be characterised on the basis of the elements of a control decision. Thereafter a detailed modelling of the processes surrounding a decision is done. This is equivalent with detailing each decision solely modelled as an activity, including the design of the decision rules.

3.2.6. Knowledge

In this 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 information objects. The important point is not the location of information usage, what has been relevant during examination of the decision processes. In contrast it has to be specified where the information objects are available in constantly updated form and thus where demanding autonomous logistic objects can access it. Examples for information locations are in turn autonomous logistic objects or can range from rather simple registers to legacy systems. For modelling the allocation of information objects knowledge maps are used. During this step the modeller may come to the conclusion that information needed for an intended decision is not available in the system. In that case the decisions have to be adapted according to the identified limitation.

3.2.7. Communication

On the basis of the processes, the decisions, their connections and the allocation of the information sources communication processes are modelled in this step. 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. The necessary communication processes are derived from the existing models. For every decision the decision maker, the necessary information objects and the information sources are determined. In a simple case the interaction protocol may only consist of a request and the related answer. More complex interaction protocols are necessary for negotiations between system elements, which arise from the connections and dependencies between decisions made by them. The object nodes modelled in activity diagrams, which only show the essential information in the first instance, are in most cases substituted by more comprehensive messages, which can also contain additional information. The design and analysis of the communication relations may lead to the conclusion that the resulting effort is too high. This conclusion may on one hand be based on the exposure of too complicated or unrealisable communication processes in this step or on simulation studies done in a later phase of the engineering process. In that case the decision processes have to be adjusted by performing the previous steps again and thus restructuring the decision processes and reallocating the information.

3.2.8. Scenario

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

3.3. FULFILMENT OF REQUIREMENTS

After presenting requirements to the modelling method earlier in this paper, this section will investigate in how far the requirements are fulfilled by the designed modelling method.

First aspect is the use of UML as the basic notation. 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 (Bauer et al, 2001), (Bauer and Odell, 2005) it is also used for knowledge modelling (Schreiber et al, 2001) or business process modelling (Oestereich et al, 2003). Its broad use makes it likely that the logistic expert assigned to the system design already came in touch with this notation earlier in one context or the other. As it is furthermore an intuitive graphical notation, with its expressiveness reduced to only the sub-set necessary here, the learning effort is accordingly low. The extensions by logistics-specific notational elements and a production logistic reference model also make the modeling method easier accessible for the logistics expert. Both of these points, the extension of the notation with logistics specifics (e.g. a layout diagram) and the reference model consisting of an ontology of production logistic concepts and an exemplary definition of autonomous objects' processes express the requirement focus on the domain of production logistics. Additionally the use of UML also fulfils the requirement of considering the later phase of software implementation. As Oestereich et al (2003) state, a language continuously used from the process model to the detailed analysis of the software system to be implemented, avoids a break in the development process, as the different fields involved all use the same semantic constructs.

Regarding the primary requirements, supporting the design of autonomous logistic objects implies an approach focused on these objects, controverts a strict top-down design approach. This will be accounted for in the procedure model by its use of a mixture of a bottom-up and top-down approach. The interacting system elements (especially the autonomous logistic objects, but also other system elements) can be shown in the Structure View. Here autonomous logistic objects can also be marked as such and their life-cycle described by an associated state-chart in the Process View. A description of the information-processing respectively of the decision processes also takes place in the Process View using Activity Diagrams. Not only an assignment of processes to the logistic objects they are located on (location of decision) is conducted here, but also the knowledge required for a decision can be modelled explicitly using object nodes. The structure of this knowledge and its initial distribution can in turn be shown in the Knowledge View, using Class Diagrams for the structure of the knowledge objects and Knowledge Maps to show its distribution. The interaction of the system elements among each other and their environment respectively is primarily described in the Communication View. UML Sequence Diagrams can be used here to specify interaction protocols. Event mechanisms (in Activity Diagrams and State Charts) can also be used to depict interaction with the environment and other system elements. They can also be used to initiate decision execution and monitor their execution progress. To be able to not only model direct communication between the autonomous logistic objects but also to allow to specify communication with the environment as a means of interaction is important to model stigmergy-based coordination. For a discussion of a stigmergy-based approach in the context of autonomous logistic processes see Scholz-Reiter et al (2006).

The heterarchical decision structure is not a characteristic of the meta-model respectively the notation, but a property of the processes in their entirety. The reference model created (not presented in this paper) has this property—there is no central entity that renders a decision which is then delegated to executing instances.

4. CONCLUSIONS

This paper addressed the topic of engineering autonomous logistic processes focussing on the modeling part. Therefore, after a short definition of autonomous control in the context of logistics, the overall system development process was sketched. After that requirements to a suitable modelling method and some relevant existing approaches were derived. The concept of our modelling method was presented subsequently, first giving a rough overview, then detailing the procedure model. The last section investigated in how far the designed modelling method fulfils the requirements derived in the beginning of the paper. We are currently working on the software tool ALEM-T supporting the notation and procedure model presented in this paper. With its help 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. Our plans for the near future are to validate our methodology on a real production logistics system.

5. ACKNOWLEDGMENT

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

Allweyer T “ModellbasiertesWissensmanagement”, *Information Management*, Vol. 13, No.1, 1998), 37–45

Bauer B, Mueller J P and Odell J, “Agent UML: A formalism for specifying multiagent software systems”, *International Journal of Software Engineering and Knowledge Engineering*, Vol. 11, No. 3, 2001, 207–230

Bauer B and Odell J, “UML 2.0 and Agents: How to Build Agent-based Systems with the new UML Standard”, *Journal of Engineering Applications of Artificial Intelligence*,Vol. 18, No. 2, 2005, 141–157

Bussmann S, Jennings N R and Wooldridge M, “Multiagent Systems for Manufacturing Control. A Design Methodology”, Springer, Berlin, Germany, 2004

Dean T and Wellman M P, “Planning and control”, Morgan Kaufmann Publishers Inc., San Francisco, CA, 1991

Finkenzeller K, “RFID-Handbook. Fundamentals and Applications in Contactless Smart Cards and Identification”, 2nd edn., Wiley, Chichester, West Sussex, 2003

Fleisch E, “Die betriebswirtschaftliche Vision des Internets der Dinge“, *Das Internet der Dinge –*

Ubiquitous Computing und RFID in der Praxis, Springer, Berlin, Germany, pp 3–38, 2005

Giret A and Botti, V, “From system requirements to holonic manufacturing system analysis”, *International Journal of Production Research*, Vol. 44, No. 18-19, 2006, pp 3917–3928

Haberfellner R, “Systems Engineering”, 11th edn, Industrielle Organisation, Zürich, Switzerland, 2002

Heinrich C, “RFID and beyond: growing your business through real world awareness”, Wiley Publishing, Indianapolis, IN, 2005

Hülsmann M and Windt, K. (Eds), “Understanding Autonomous Cooperation & Control in Logistics – The Impact on Management, Information and Communication and Material Flow”, Berlin, Springer, 2007

Laux H, “Entscheidungstheorie”, Springer, Berlin, Germany, 2005

Malone T W and Crowston K, “The Interdisciplinary Study of Coordination”, *ACM Computing Surveys*, Vol. 26, No. 1, 1994, pp 87–119

McFarlance D C and Bussmann S, “Holonic manufacturing control: rationales, developments and open issues”, *Agent-Based Manufacturing – Advances in the Holonic Approach*, Springer, Berlin, 2003, pp 303–326

Monostori L, Vancza J and Kumara S R T, ”Agent-Based Systems for Manufacturing”, *CIRP Annals*, Vol. 2, 2006, pp 697-720

Oestereich B, Weiss C, Schröder C, Weilkiens T and Lenhard, A, “Objektorientierte Geschäftsprozessmodellierung mit der UML”, Dpunkt, Heidelberg, Germany, 2003

Oestereich B, „Objektorientierte Geschäftsprozessmodellierung und modellgetriebene Softwareentwicklung“, *HMD - Praxis der Wirtschaftsinformatik*, Vol. 41, No. 241, 2005, pp 27–34

OMG, “Unified Modeling Language Specification. Version 2.0” Retrieved: 01 December 2006, www.uml.org

Patig S, “Flexible Produktionsfeinplanung mit Hilfe von Planungsschritten: ein Planungsansatz zum Umgang mit Störungen bei der Produktion“, Peter Lang, Frankfurt a.M., Germany, 2001

Scheer A-W, „ARIS - Modellierungsmethoden, Metamodelle, Anwendungen“, 4th edn., Springer, Berlin, 2001

Scholz-Reiter B, Windt K and Freitag M, “Autonomous logistic processes: New demands and first approaches”, *Proceedings of 37th CIRP International Seminar on Manufacturing Systems*, Hungarian Academy of Science, Budapest, Hungaria, 2004, pp 357-362

Scholz-Reiter B, Freitag M, de Beer C and Jagalski T, “Modeling and simulation of pheromone based shop floor control”, *Proceedings of the 3rd CIRP Conference Digital Enterprise Technology*, Escola, Setubal, Portugal, 2006

Schreiber G, Akkermans H, Anjwerden A, de Hoog R, hadbolt N, van de Velde W and Wielinga B, “Knowledge Engineering and Management - The CommonKADS Methodology”, 3rd edn., MIT Press, Cambridge, MA, 2001

Sowa J F, “Knowledge representation: logical, philosophical, and computational foundations”, Brooks/Cole Publishing Co., Pacific Grove, CA, 2000

Valckenaers P, Heikkilä, T, Baumgaertel H, McFarlane D and Courtois J-P, “Towards a novel manufacturing control principle”, *Proceedings of the Second International Workshop on Intelligent Manufacturing Systems*, Katholieke Universiteit Leuven, 1999, pp 871–875

Weiss G, “Multiagent Systems - A Modern Approach to Distributed Artificial Intelligence”, 2nd edn., MIT Press, Cambridge, MA, 2000

Weiss G and Jakob R, “Agentenorientierte Softwareentwicklung”, Springer, Berlin Heidelberg, Germany, 2005

Windt K, “Selbststeuerung intelligenter Objekte in der Logistik“, *Selbstorganisation – Ein Denksystem für Natur und Gesellschaft*, Köln, Böhlau, 2006

Wooldridge M and Jennings N R, “Intelligent agents: theory and practice”, *The Knowledge Engineering Review*, Vol. 2, No. 10, 1995, pp 115–152