

Engineering autonomously controlled logistic systems

B. Scholz-Reiter, J. Kolditz* and T. Hildebrandt

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

(Received 30 January 2007; final version received 16 July 2007)

Today enterprises are exposed to an increasingly dynamic environment. Last but not least increasing competition caused by globalization 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 logistic 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 will be able to specify an autonomous logistic system adequately. Further research will later on complement the methodology to support the whole engineering process.

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, for example, by globalisation and the exposure of enterprises to an increasingly dynamic environment. One possibility when facing increasing dynamics is autonomous control of logistic processes. This will allow more robust processes in spite of growing environmental as well as internal complexity.

Autonomous control describes processes of decentralised decision-making in heterarchical structures. It presumes interacting elements in non-deterministic systems, which possess the capability and possibility to render decisions independently. The objective of autonomous control is the achievement of increased robustness and positive emergence of the total system due to distribution and flexibility coping with dynamics and complexity (Hülsmann and Windt 2007). As the focus lies in the areas of production and transport logistics the system elements making their decision autonomously are the logistic objects such as commodities, machines, storages and conveyors themselves (Scholz-Reiter *et al.* 2006b).

*Corresponding author. Email: kol@biba.uni-bremen.de

In order to enable logistic objects to be 'intelligent' they have to be provided with smart labels. Today's RFID (radio frequency identification) chips have very limited capabilities with respect to energy, range, storage capacity and especially information processing (Finkenzeller 2003). Nowadays RFID is already widely used in industry for identification matters and several ideas for future applications exist (Das and Harrop 2001, Fleisch 2005, Heinrich 2005). Soon highly evolved smart labels will provide resources like micro computers to logistic objects, allowing the 'pure vision' of autonomous logistic processes to be realised. In contrast to manufacturing execution systems (MES) that are also meant to support manufacturing control better than enterprise resource planning (ERP) based systems, autonomous control presumes heterarchical systems and allocates intelligence to the logistic objects to obtain a logical as well as physical distribution of decision making.

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?', 'What logistic objects are there?' and 'How much "intelligence" do they have?' This information is the basis on which to create a model for a suitable control strategy; including, for instance, the processes on the autonomous logistic objects, what knowledge is required for these processes and the communication or other co-ordination mechanisms between the objects.

This paper presents the engineering framework ALEM (autonomous logistics engineering methodology) and focuses on a modelling concept as part of it, and is structured as follows. Section 2 presents the overall systems engineering framework. Section 3 discusses process modelling under the paradigm of autonomy. Starting with requirements the section introduces the modelling methodology and details important aspects of it. Thereafter exemplary models are explained. The paper concludes with a short summary and a look at future work.

2. Engineering an autonomous logistic systems

The engineering of an autonomous logistic 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 single phases of the engineering process are described and the autonomous logistics engineering methodology (ALEM) is introduced. The focus of ALEM is methodological support of the phases of 'main study' and 'detailed studies' of the general procedure.

1. *Initiation*: The rather unstructured initiation phase is triggered by sensing a problem and is completed by the decision to start a preliminary study. In our context this might be a problem associated with production planning and control or the assumption of an opportunity to improve the logistic system's performance by adoption of autonomous control.
2. *Preliminary study*: During the preliminary study phase the objectives of adopting autonomous control have to be defined aimed at an improvement in the fulfilment of logistic goals. In this regard the considered system and the scope of work have to be stated; for example, a certain area of a production system. Part of the preliminary study is also a situational analysis, which provides an overall understanding of the scope of work, of existing problems and control processes. If required this system analysis can be detailed for certain aspects in later phases.

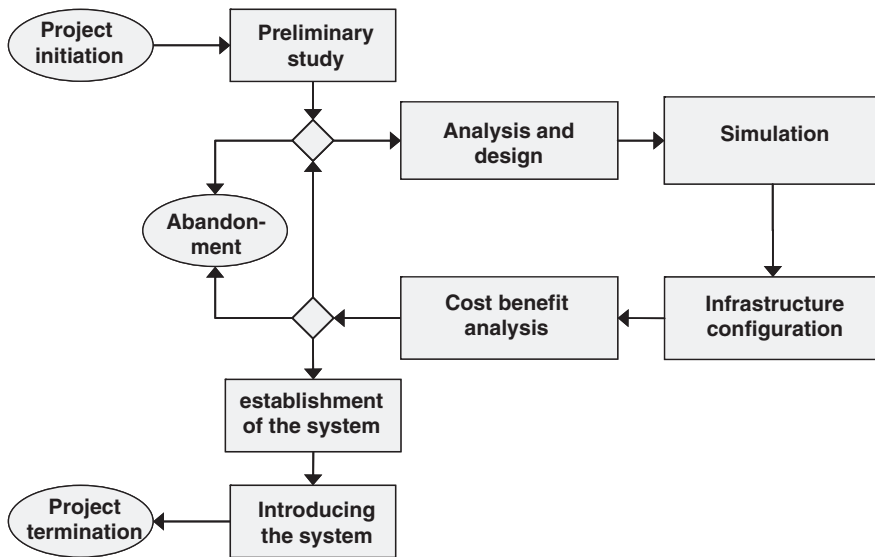


Figure 1. Concretised systems engineering procedure model for autonomous logistic systems (in extension of Haberfellner 2002).

An important basis for the decision whether to continue or to abandon the project is an estimation of the impact of the solution principle. Therefore it is estimated to what extent an application of autonomous control is reasonable and promising. This allows a decision whether to start the main study phase or to cancel the project in respect to the development of an autonomous system. The comparison of autonomous control and alternative methods, and therewith the rating of such a solution principle under certain conditions, is an issue of ongoing research that will not be discussed here any further. For more details see Philipp *et al.* (2007).

3. *Analysis and design phase*: These two core-phases of the engineering process are seen as an iterative process in ALEM, as shown on the right in Figure 1.

Methodological support of this iterative process for the development of an autonomous logistic system is the focus of ALEM.

- 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 pro-active elements in the 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 are 'intelligent' in respect of 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 and software engineering the design created before is first tested in a simulation. Operability and impact on logistics performance of the whole system are of special concern here. A central task is the verification of required system behaviour, because this is a necessary precondition for industrial application of emergent systems such as

autonomous logistic systems. Moreover, this step also allows an effective comparison of existing or alternative ways of controlling the logistical 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, more detail is required with every iteration loop. Conclusions may be drawn from the process model as well as from the simulation. For example, allocation of control processes and data packets plus 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 the case of repeating negative results in this step an application of autonomous control has to be abandoned for this scenario.
4. *Establishment of the system*: During the establishment step the autonomous logistic system is realised. The main topics are the software implementation and the creation and integration of facilities and instruments. Ideally the software should be implemented using parts of the program code created during the simulation step.
 5. *Introduction of the system and termination of the project*: Normally the introduction will involve huge and complex systems resulting in hardly any, or even incalculable, side effects. Therefore the introduction of the autonomous logistic system should be done stepwise if possible. After verifying the fulfilment of the objectives the system is handed over from the originating project team to the operating institution and the engineering project is terminated.

3. Modelling autonomous control

This section concentrates on the first step of the modelling cycle, the semi-formal specification of the autonomous logistic system. 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. 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.

The remainder of this section is organised as follows: First presenting requirements to the modelling methodology and existing approaches are evaluated concerning these requirements, ALEM-N and ALEM-P are described in the main part of the section, after which the fulfilment of the requirements by ALEM-N and -P are investigated.

3.1 Requirements to the modelling methodology and existing approaches

Formulating requirements for a methodology for modelling autonomous logistic processes has to start on the basis that it has to meet the definition of a modelling methodology in general. A modelling methodology is a systematic approach which defines the essential modelling-related 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, such as the 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 the one hand, assure realization of decisions of logistic objects in the material flow and, on the other hand, to allow the monitoring of progress by observing the environment.

In a heterarchically structured system of autonomous elements, such as an autonomous logistic system, intensive interaction of the system elements is required to co-ordinate their actions, which results in the high importance of communication (Malone 1994, Weiss 2000). An additional requirement for the methodology is therefore the possibility of modelling the communication of the system elements.

The task of constructing a model of autonomous logistic processes shall primarily be assigned to an expert for the 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 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, 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 (McFarlane 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 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. In the context of software engineering, methodologies for business process modelling are intended to support the development of centralised information systems (Scheer 2000). Due to this purpose dedicated concepts for specifying decentralised approaches, instruments for detailed illustration of local information processing, techniques for explicit modelling of communication processes and protocols and particularly a procedure supporting the modeller in creating a system based on autonomous logistic processes instead of a centralised information system, 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 extent 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.

3.2 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 2000) which is

also reflected in the guideline of systematic design (see the last paragraph of the subsection on requirements). 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 as 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. Most UML models used in logistics focus on static aspects of centralized planning and control systems (Kees 1998) and therefore cannot be used for autonomous systems. Nevertheless they should be used as a modelling basis and be adapted during specification of the structural part model.

The 'knowledge view' describes the knowledge, which has to be present in the logistic objects, to allow decentralised 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, while for the just-mentioned temporal aspects, a dedicated knowledge representation language would have to be used (Sowa 2000). 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 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 depict the interaction of autonomously deciding, otherwise only loosely coupled objects 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. 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 extent, it shows an external view onto the system, its elements and their relations and interactions. On the other hand, 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. Nevertheless not all views use both perspectives to the same extent.

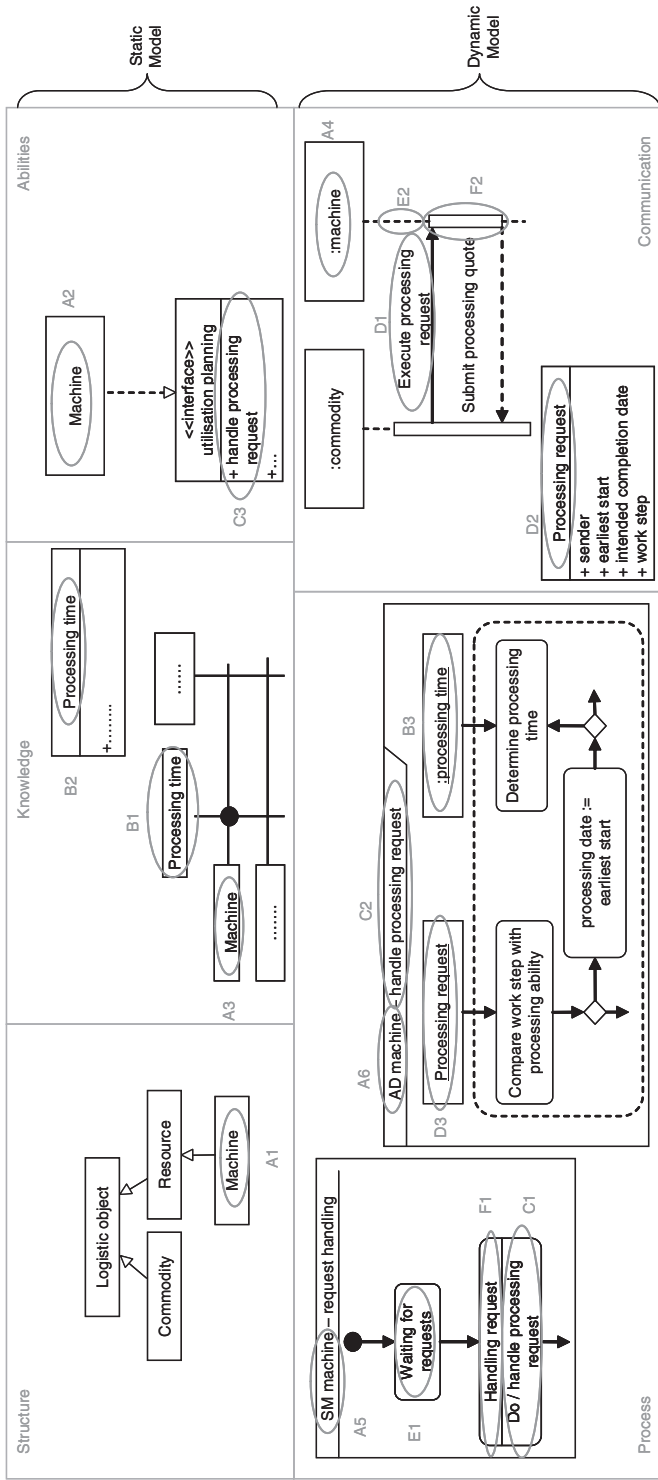


Figure 2. ALEM-N—connections of views and diagrams.

Also illustrated in Figure 2 are the connections between the views. The connections 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. The correlated items are named by a shared letter as well as a digit that allows an individual reference.

In the structure view the system elements are defined, which are used and explained in more aspects in the other views. That makes this view the central and connecting element of the view concept. Thus the system elements modelled as classes in the structure view (A1) correspond to the ones matched with abilities in the ability view (A2). In the knowledge view system elements (A3) are determined as locations for certain information objects. The communication partners appearing in the communication view (A4) are instances of classes from the structure view and processes defined in the process view (A5, A6) are attached to system elements modelled in the structure view.

States between messages (E2, F2) of autonomous logistic objects modelled in the communication view (A4) and transitions caused by incoming or outgoing messages are explained in more detail in state machines of the process view (E1, F1). Furthermore activities that have to be performed in the states (C1) may be explained in activity diagrams (C2), which causes a connection between different diagrams within the process view (C1–C2). The messages appearing in sequence diagrams of the communication view (D1) may be described in the same view by class diagrams in more detail (D2). Instances of these communication objects in turn may occur as object nodes in the process view (D3). Object nodes in the process view (B3) may alternatively be instances of information objects that are defined in the knowledge view (B2) and attached (B1) to other system elements. Between process and ability view there is a connection by the processes (C2) needed to realise certain abilities (C3).

3.3 *ALEM-P: the procedure model*

The procedure model is a guideline for modelling autonomous logistic processes, which contributes on the 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 detailed knowledge of logistics planning and control is able 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, and then activities to perform and results to get out of every step. Furthermore methods and instruments are recommended to support the work. Among these are, first, the presented view concept and diagrams; second, modelling conventions in terms of construction and consistency rules and, third, 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 re-engineering 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 finalised. However, the focus on selected autonomous logistic objects and their reciprocal co-ordination with each other, as well as the other system elements, shows the 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.

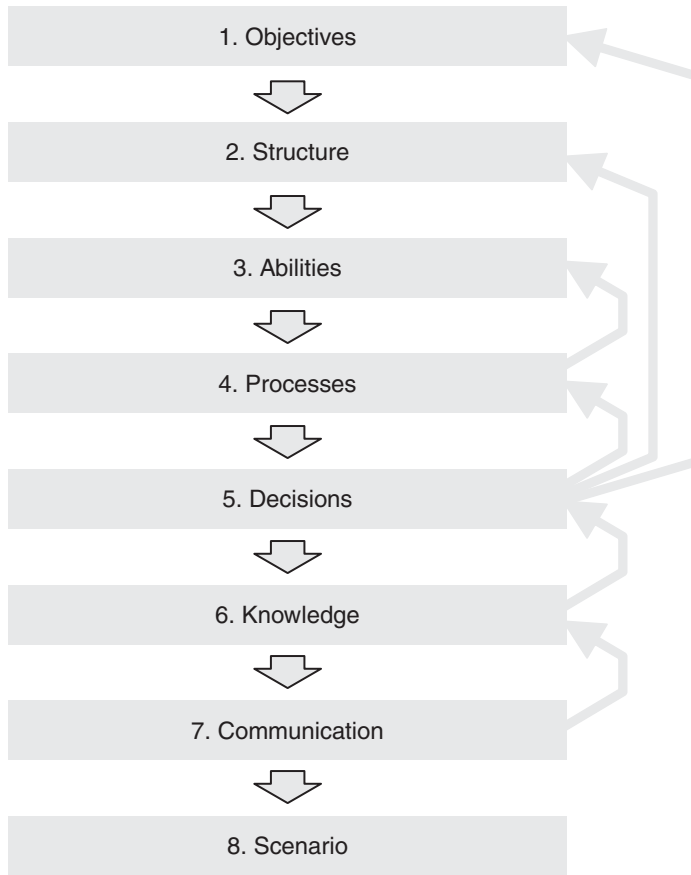


Figure 3. ALEM-P—procedure model.

3.3.1 Objectives

The first step in the specification procedure for autonomous logistic systems broaches the issue of objectives in the system. The starting point is the global system objectives that have to be clarified by the modeller, if needed, in cooperation with the person in charge of strategic topics. For a production system the classic goals of production logistics will 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.

3.3.2 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 will 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.

3.3.3 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 the 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 an increasingly 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 the one hand the agenda for the following process design is set, and on the other hand there may have to be adjustments in the mapping and structuring according to subsequent findings.

3.3.4 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 then 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.

At 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 progress 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 of an individual object. 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 one to initially disregard the necessary information acquisition and thus 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 logistic 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. The disturbances have to be especially 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, disturbances which need human intervention also have to be taken into account, for example to be able to provide adequate human-machine interfaces.

3.3.5 *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 to detailing each decision solely modelled as an activity, including the design of the decision rules.

3.3.6 *Knowledge*

In this step the focus lies on the knowledge needed for decision making. For that purpose every decision has to be analysed as to 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, but 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. Examples for knowledge 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.

3.3.7 *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 the 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. The communication processes result from the logical and temporal configuration of the processes embedding the decisions. 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 on their part contain amongst others the essential information.

The design and analysis of the communication relations may lead to the conclusion that the resulting effort is too high. This conclusion may 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.

3.3.8 *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 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.

3.4 *Fulfilment of requirements*

After presenting requirements to the modelling method from earlier in this paper, this section will investigate how far the requirements are fulfilled by the designed modelling method as presented in the previous part of this paper. First of all the fulfilment of the two general, i.e. non-functional, requirements of a focus on the domain of production logistics and the logistic expert as a modeller will be investigated. The latter requirement can, for instance, be found in the use of UML as the basis of the modelling notation used. As a graphical, semi-formal notation it is broadly used—besides software development (agent-oriented approaches are of particular interest here; see, for instance, AUML (Bauer *et al.* 2001, Odell *et al.* 2001)) 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 process also make the 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 process consisting of 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, objecting 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 co-ordination (for a discussion of a stigmergy-based approach in the context of autonomous logistic processes, see Scholz-Reiter *et al.* 2006). Stigmergy means indirect decentralized information flows that occur when peers make and sense changes to environmental variables, what is common for example in insect colonies (Parunak and Brueckner 2004).

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 has this property—there is no central entity that renders a decision which is then delegated to executing instances.

4. Exemplary model

This section will give a short overview to the application of the modelling methodology on the basis of a real case we use to evaluate the modelling methodology presented concerning adequate assistance during modelling and later on concerning transformation of the generated models into an agent-based simulation. The real case is a factory of an automotive supplier for engine and gear components. We focus on the production of valve spring holders which are manufactured in a job shop production in several stages with a

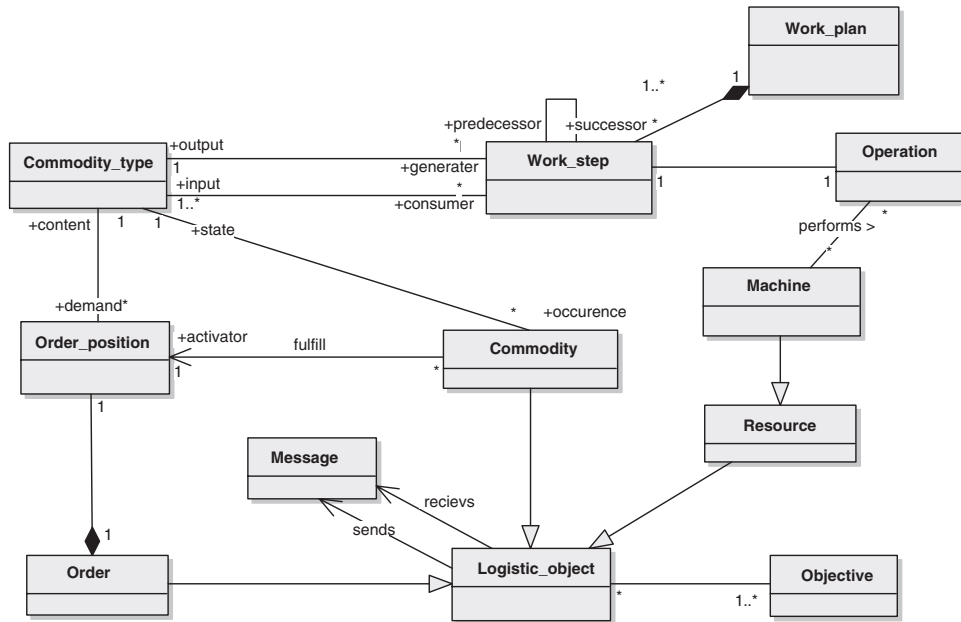


Figure 4. Class diagram showing a part of the taxonomy supporting the user and selected relationships between the classes shown.

varying number of alternative machines on each stage. Valve spring holders are a matter of bulk material and are transported in lattice boxes on the shop floor.

Corresponding to the procedure model ALEM-P first the objectives for the production system have to be defined. The production logistic objectives of short throughput times, low work-in-progress level, high utilisation and high delivery reliability (Nyhuis *et al.* 2005) are also used for the autonomous logistic processes. Finalization of these global objectives for individual logistic objects will be done later. Building the structure of the autonomous logistic system, machines and commodities are interpreted as autonomous logistic objects. However a commodity is not meant to be a single valve spring holder, but a lattice box filled with an amount of products being the minimal lot size. For simplification reasons the transportation between machines and stockyards using forklifts is disregarded here just as operators are. This structure is shown in Figure 4.

The standard processes and those considering uncertainties are modelled in step four. For a commodity three main states are differentiated as shown in the left part of Figure 5: 'searching', 'waiting' and 'in process'. If the scheduled machine breaks down during waiting and thereby the processing date is delayed for an indefinite period of time, the commodity state is changed from 'waiting' to 'searching' instead of 'in process'. In this state a machine is selected again according to the activity diagram shown in the right part of Figure 5.

In that activity diagram the action 'select machine' can be identified as a decision. According to the decision model to be used the decider is 'commodity', trigger is the activity 'identify possible machines' and the alternatives are 'machines'. Now the global objectives have to be finalised referring to step 1. For the commodity the objective 'short time until completion of work step' may be derived, what means for the decision model,

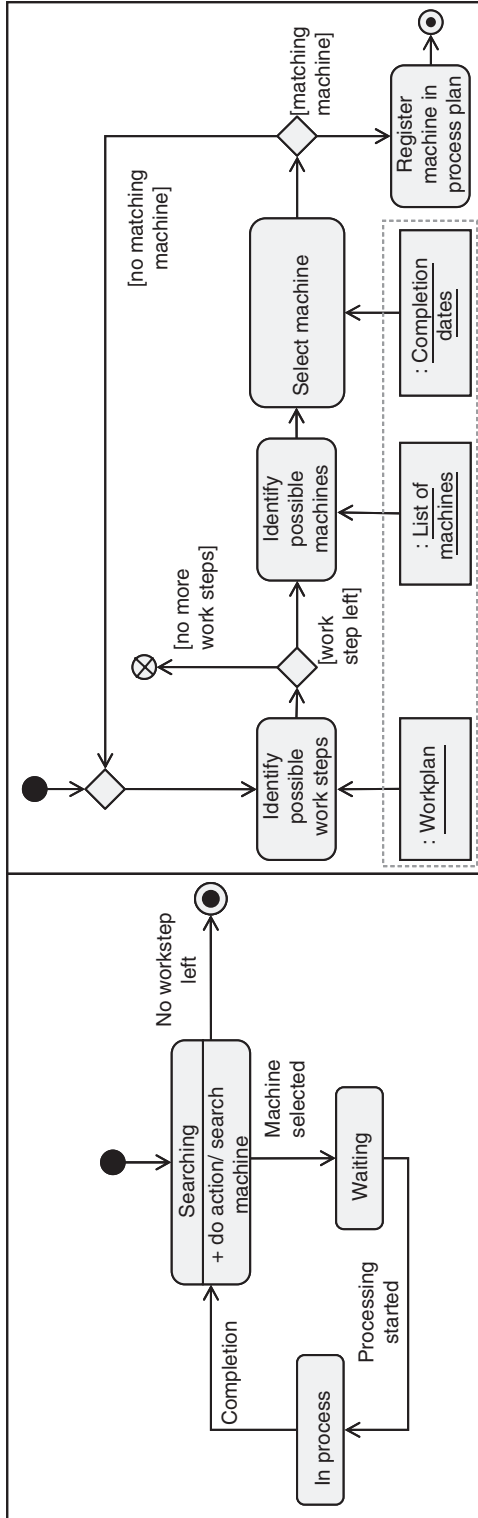


Figure 5. State machine of commodity and activity diagram *search machine*.

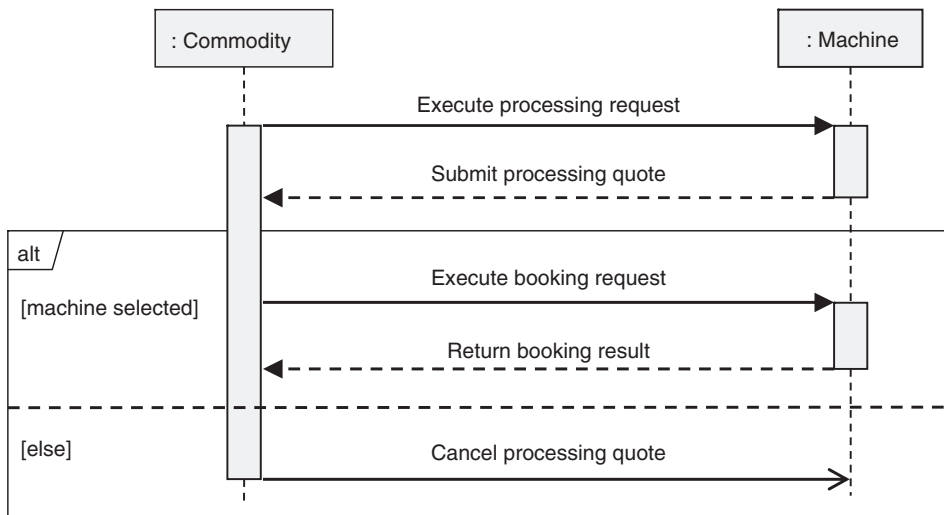


Figure 6. Sequence diagram machine selection.

that the effects connected to the alternatives are the respective 'completion dates'. Hence the decision rule 'choose earliest completion date' is defined.

During step six, 'knowledge', the object nodes are added in activity diagrams (see dashed area in Figure 5) and knowledge objects are allocated using knowledge maps. The commodity needs details about next possible work steps, about machines to be considered and the feasible completion dates. While the work plan is assigned to the commodity the possible machines and completion dates will come from the machines themselves.

The communication between a commodity and a machine is modelled in step seven of the procedure model. Figure 6 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. After the commodity has selected a machine on basis of the decision specified above the chosen machine is booked by the commodity. The other machines are informed about the quote cancellation. In Figure 6 this is modelled by a combined fragment of the type 'alternative'.

In step eight the instances existing in the production system are acquired. On one hand there are stamping machines, ovens, bonding machines and washing machines. On the other hand there are about 100 variants of valve spring holders with their individual alternative work plans and different possible manufacturing states. The machines are created in the layout diagram used in the structure view; other instances are integrated by using object diagrams or lists.

5. Summary

This paper addressed the topic of engineering autonomous logistic processes focusing on the specification as part of the analysis and design phase. 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 were derived. The concept of our modelling method was presented subsequently, first giving a rough overview, then detailing the view concept, a part of the notation and the procedure model. After that description we investigated as to how far the designed modelling method fulfils the requirements derived at the beginning of the paper. In the last section selected aspects of modelling a real case from the domain of production logistics were presented.

We are currently working on the software tool ALEM-T (Scholz-Reiter *et al.* 2007) supporting the notation and procedure model presented in this paper. With its help a process expert (e.g. a logistics expert with only a little background in computer science) will be supported in modelling and designing autonomous logistic processes. We are further validating our methodology with the sketched production logistic system and a transformation of the model into an agent-based simulation. Our plans for the future are to expand ALEM for full methodological and tool support for the iterative modelling process consisting of specification, simulation, hardware configuration and cost benefit analysis.

Acknowledgement

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. and Odell, J., 2001. Agent UML: A formalism for specifying multiagent software systems. *International Journal of Software Engineering & Knowledge Engineering*, 11 (3), 207–230.
- Bauer, B. and Odell, J., 2005. UML 2.0 and Agents: How to Build Agent-based Systems with the new UML Standard. *Journal of Engineering Application of Artificial Intelligence*, 18 (2), 141–157.
- Bussmann, S., Jennings, N.R. and Wooldridge, M., 2004. *Multiagent systems for manufacturing control. A design methodology*. Berlin, Germany: Springer.
- Das, R. and Harrop, P., 2001. *The internet of things: massive new markets for automated location, tracking, authentication and barcode replacement. total asset visibility*. Cambridge, UK: IDTechEx.
- Dean, T. and Wellman, M.P., 1991. *Planning and control*. San Francisco, CA: Morgan Kaufmann Publishers Inc.
- Finkenzeller, K., 2003. *RFID Handbook*, 2nd ed. Chichester, West Sussex, England: John Wiley.
- Fleisch, E., 2005. Die betriebswirtschaftliche Vision des Internets der Dinge. In: E. Fleisch and F. Mattern, eds. *Das internet der dinge—ubiquitous computing und RFID in der Praxis* Berlin, Germany: Springer, 3–38.
- Giret, A. and 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*, 11th ed. Zürich, Switzerland: Industrielle Organisation.
- Heinrich, C., 2005. *RFID and beyond: Growing your business through real world awareness*. Indianapolis, IN: Wiley Publishing.
- Kees, A., 1998. *Ein verfahren zur objektorientierten modellierung der produktionsplanung und -steuerung*. Aachen, Germany: Shaker.

- Hülsmann, M. and Windt, K. (Eds), 2007. *Understanding Autonomous Cooperation & Control in Logistics—The Impact on Management, Information and Communication and Material Flow*. Berlin, Germany: Springer.
- Kotonya, G. and Sommerville, I., 1998. *Requirements engineering*. Chichester, West Sussex, England: John Wiley.
- Laux, H., 2005. *Entscheidungstheorie*. Berlin, Germany: Springer.
- Malone, T.W. and Crowston, K., 1994. The interdisciplinary study of coordination. *ACM Computing Surveys*, 26 (1), 87–119.
- McFarlane, D.C. and 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.
- Monostori, L., Vancza, K. and Kumara, S.R.T., 2006. Agent-based systems for manufacturing. *CIRP Annals*, 55 (2), 697–720.
- Nyhuis, P., von Cieminski, G. and Fischer, A., 2005. Applying simulation and analytical models for logistic performance prediction. *CIRP Annals*, 54 (1), 417–422.
- Odell, J.J., Parunak, H.V.D. and Bauer, B., 2001. Representing agent interaction protocols in UML. In: P. Ciancarini and M. Wooldridge, eds. *Agent oriented software engineering*. Berlin, Germany: Springer, 121–140.
- Oestereich, B., Weiss, C., Schröder, C., Weikiens, T. and Lenhard, A., 2003. *Objektorientierte geschäftsprozessmodellierung mit der UML*. Heidelberg, Germany: Dpunkt.
- OMG, Unified Modeling Language Specification, Version 2.0. Available online at: www.uml.org (Accessed 1 December 2006).
- Parunak, H.V.D. and Brueckner, S.A., 2004. Engineering swarming systems. In: F. Bergenti, M.-P. Gleizes and F. Zambonelli, eds. *Methodologies and software engineering for agent systems: the agent-oriented software engineering handbook*. Berlin, Germany: Springer, 341–376.
- Patig, S., 2001. *Flexible produktionsfeinplanung mit hilfe von planungsschritten: ein planungsansatz zum umgang mit störungen bei der produktion*. Frankfurt a. M., Germany: Peter Lang.
- Philipp, T., de Beer, C., Windt, K. and Scholz-Reiter, B., 2007. Evaluation of autonomous logistic processes—Analysis of the influence of structural complexity. In: M. Hülsmann and K. Windt, eds. *Understanding autonomous cooperation and control in logistics—the impact on management, information and communication and material flow*. Berlin, Germany: Springer, 303–324.
- Scheer, A.-W., 2000. *ARIS—Business process modeling*. Berlin, Germany: Springer.
- Schönsleben, P., 2001. *Integrales informationsmanagement: informationssysteme für geschäftsprozesse—management, modellierung, Lebenszyklus und Technologie*, 2nd ed. Berlin, Germany: Springer.
- Scholz-Reiter, B., Freitag, M., de Beer, C. and Jagalski, T., 2006. Modeling and simulation of pheromone based shop floor control. In: P.F. Cunha and P. Maropoulos, eds. *Proceedings of the 3rd CIRP conference digital enterprise technology*, [CD-ROM]. Setubal, Portugal: Escola.
- Scholz-Reiter, B., Windt, K., Kolditz, J., Böse, F., Hildebrandt, T., Phillip, T. and Höhns, H., 2006. New concepts of modelling and evaluating autonomous logistic processes. In: G. Chrysolouris and D. Mourtzis, eds. *Manufacturing, modelling, management and control 2004*. Amsterdam, The Netherlands: Elsevier Science.
- Scholz-Reiter, B., Hildebrandt, T. and Kolditz, J., ALEM-T: 2007. A Modelling Tool for Autonomous Logistic Processes. In: B. Hon and P. Maropoulos, eds. *Proceedings of 40th CIRP seminar on manufacturing systems*, [CD-ROM]. Department of Engineering, University of Liverpool: Liverpool, UK.
- Schreiber, G., Akkermans, H., Anjwerden, A., de Hoog, R., Shadbolt, N., van de Velde, W. and Wielinga, B., 2001. *Knowledge engineering and management—the commonKADS methodology*, 3rd ed. Cambridge, MA: MIT Press.
- Schuette, R. and Rotthowe, T., 1998. The guidelines of modelling as an approach to enhance the quality of information models. In: T.W. Ling, S. Ram and M.L. Lee, eds. *Proceedings of the 17th international conference on conceptual modeling*, Berlin, Germany: Springer, 240–254.
- Sowa, J. F., 2000. *Knowledge representation: logical, philosophical, and computational foundations*. Pacific Grove, CA: Brooks/Cole Publishing Co.

- Valckenaers, P., Heikkilä, T., Baumgaertel, H., McFarlane, D. and Courtois, J.-P., 1999. Towards a novel manufacturing control principle. In: H. Van Brussel and P. Valckenaers eds., *Proceedings of the second international workshop on intelligent manufacturing systems*. Leuven, Belgium: Katholieke Universiteit Leuven, 871–875.
- Weiss, G., 2000. *Multiagent systems—a modern approach to distributed artificial intelligence*, 2nd ed. Cambridge, MA: MIT Press.
- Windt, K., 2006. Selbststeuerung intelligenter Objekte in der Logistik. In: M. Vec, M.-T. Hütt and A.M. Freund, eds. *Selbstorganisation—ein denkssystem für natur und gesellschaft*. Cologne, Germany: Boehlau, 271–314.
- Wooldridge, M. and Jennings, N.R., 1995. Intelligent agents: theory and practice. *The Knowledge of Engineering Review*, 2 (10), 115–152.