An Ontology for Production Control of Semiconductor Manufacturing Processes

Lars Mönch and Marcel Stehli

Technical University of Ilmenau,
Institute of Information Systems,
Helmholtzplatz 3, 98684 Ilmenau, Germany
{Lars.Moench,Marcel.Stehli}@tu-ilmenau.de

Abstract. In this paper, we describe an ontology for a hierarchically organized production control system in semiconductor manufacturing. The semiconductor manufacturing domain is characterized by reentrant product flows, sequence dependent setup-times, prescribed due-dates, a diverse product mix, a mix of different process types including batch processes and preventive maintenance issues because of complex technological processes. Starting from a hierarchical decomposition of the manufacturing system, we use an agent-based architecture for implementing the resulting production control system. In order to coordinate the autonomous entities of the hierarchy, we suggest an ontology that is appropriate to the hierarchical control approach. We illustrate the use of the suggested ontology.

1 Introduction

A complex system is defined as a system whose components are systems in their own right, i.e., a complex system is a relation of interconnected sub-systems (cf. [7] for a formal description of this concept). In this paper, we use the term complex manufacturing system for systems that can be decomposed into a set of manufacturing subsystems that have their own goals and rights for decision-making.

We consider complex manufacturing systems that are also characterized by

- a diverse product mix,
- reentrant process flows,
- parallel machines with complicated dedication practices,
- sequence-dependent setup-times,
- a mix of different process types, including batch processes,
- inclusion of preventive maintenance issues into production control,
- prescribed due-dates for the lots,
- concurrence of production and engineering lots for scarce machine capacities.

Examples of this type of complex manufacturing systems are semiconductor wafer fabrication facilities (wafer fabs). The development of efficient planning and control strategies is highly desirable in the semiconductor manufacturing domain, because the improvement of operational processes creates the best opportunity to realize the nec-

M. Schillo et al. (Eds.): MATES 2003, LNAI 2831, pp. 156–167, 2003.
© Springer-Verlag Berlin Heidelberg 2003

essary cost reductions. For a more detailed description of the semiconductor manufacturing domain and related production control issues we refer to [8].

We describe an ontology that takes the hierarchical structure of the control system (and, of course, the resulting multi-agent-system) into account. The ontology is designated for the implementation of a content language in the multi-agent-system FABMAS (cf. [2], [10] for the architecture of the FABMAS system). Ontologies for the manufacturing domain have been revisited by many authors (cf. [13], [16], [11]). However, so far an ontology for a hierarchically organized production control system in the semiconductor manufacturing domain has not developed yet.

This paper is organized as follows. In Section 2, we describe the manufacturing domain that is relevant for our research. We also give an outline of the hierarchical production control approach including an agentification. In Section 3, we develop an ontology for hierarchically organized production control systems. We finish the paper by describing the use of the ontology by means of an example.

2 Hierarchical Production Control Approach

We consider a manufacturing system that consists of different work areas. Each of the work areas includes several groups of parallel machines (called work center). The work areas can organize their work in an autonomous way, i.e., there is only a very weak form of central control. Based on the described physical decomposition of the manufacturing system, we suggest a hierarchical multi-layer approach (cf. [7], [12] for the general theory behind this type of approach).

We consider the full wafer fabrication facility as an upper level of the hierarchy. We use a beam-search-type algorithm (cf. [5]) for minimizing the deviation of the completion time of the lots from their planned due date. As a result of the algorithm, we obtain start and end dates for the lots for each single work area.

The middle layer is formed by the different work areas. We use an area-based shifting bottleneck type algorithm (ABSBH) in order to minimize the total weighted tardiness for each single work area. We measure the tardiness with respect to the work area-related start and end dates that are given by the upper level. This layer results in schedules for each single work area. Note that we can treat each single work area separately because of the decoupling effect of the beam-search algorithm.

The lower level is basically formed by work centers. The work centers should implement the schedules obtained from the mid level. In the case of invalid schedules we suggest the use of a modified contract net with properly chosen costs (cf. [9] for the implementation of a machine hour-rate (MHR)-based resource allocation algorithm).

The solution of the decision-making problems on the three levels requires communication and intra- and inter-layer coordination across the decision-making units. For example, it is required to send information on machine break-downs from the manufacturing process up to the upper level in order to update the used models for decision-making. Agents are an appropriate way of implementing the described production control approach because, by definition, agents are decision-making entities. The agents need to communicate. The key elements of an agent communication language are suggested by FIPA. However, it is not possible to specify the content of such a

language in a generic way, because it is highly domain- and task-dependent. An appropriate ontology is a prerequisite for the development of a content language for hierarchically organized complex manufacturing systems.

Starting with the PROSA architecture [17], we distinguish between decision-making agents and staff agents. Decision-making agents solve decision problems, while the staff agents support them in the course of the decision-making process. We identify seven decision-making agent types in our application scenario. It is summarized in Table 1.

Table 1. Agentification Due to PROSA Reference Architecture

Agent Type	Description
Lot agent	- represents a single lot
PM agent	- represents a preventive maintenance order
Tool agent	- represents a certain auxiliary resource
Work center agent	- represents a single group of parallel machines on the shop floor
Work area agent	- represents a set of work centers
Fab agent	- represents all work areas
Technology agent	- represents the product knowledge

We also define different staff agents that encapsulate scheduling and monitoring functionality. In Table 2, we summarize the basic functionality of the members of the agency that corresponds to the different hierarchy levels. For a more detailed description we refer to [8].

3 Ontology for Production Control of Wafer Fabs

We define an ontology as a model of a particular field of knowledge, the concepts and their attributes, as well as the relationships between the concepts [4]. An ontology always includes always

1. a conceptualization of a domain, i.e. how to view and model a domain,
2. a specification of this conceptualization, i.e. basically a formal description.

Both conceptualization of the domain and specification of the conceptualization are influenced by the used modeling technique, i.e. slots or frames [11]. In this paper, we describe an ontology for a hierarchically organized multi-agent-system. As pointed out in [1], we have to describe the underlying predicates, i.e. expressions that can be true or false, and also the agent actions beside the domain concepts. Following this point of view, we have to develop an application ontology that contains both domain- and task-related parts (cf. [3] for a classification of ontologies).

Table 2. Functionality of the Members of the Agency (Fab, Work Area, Work Center Level)

Layer of the Hierarchy	Corresponding Member of the Agency	Task Description
Entire Fab	Fab Agent	- coordination of the work of the fab scheduling agent, the monitoring agent and the work area agents - decision-making through sequencing the lots for the application of the beam-search algorithm
	Fab Scheduling Agent	- preparation of beam-search algorithm run - running the algorithm - providing scheduling information
Work Area	Work Area Agent	- coordination of the work of the corresponding work area scheduling and monitoring agent - decision-making through choosing the proper machine criticality measure for ABSBH - information providing services
	Work Area Scheduling Agent	- preparation of ABSBH algorithm run - running the heuristic - providing scheduling information
Work Center	Work Center Agent	- implementing the work area schedules in a dispatching manner - works as mediator in case of the contract-net-type allocation algorithm

The ontology to be developed should support scheduling, rescheduling and control activities in a hierarchically organized multi-agent-system. In particular, it should allow for interactions between the agents that represent the units of the manufacturing system in the following situations:

1. Treatment of exceptions,
2. Start of scheduling activities,
3. Exchange of information that is required for scheduling,
4. Transfer of scheduling results back to the decision-making agents,
5. Start of monitoring activities,
6. Transfer of monitoring results back to the decision-making agents,
7. Negotiations between lot agents and machine agents via mediator,
8. Exchange of data and information that is required for distributed scheduling according to the ABSBH algorithm.

To develop an appropriate ontology, we start from a data model suitable for the treatment of this type of manufacturing processes. We have to distinguish between static data, describing the shop-floor and products, dynamic data, i.e. the state of lots and single machines, and various kinds of control data, i.e. schedules and information on certain events on the shop-floor. A data model must therefore contain the data given in Table 3.

The basic data model (without control data) is shown in UML (Unified Modeling Language) notation in Figure 1. Here, we describe the association of the data by using

Table 3. Data Required in the Semiconductor Manufacturing Domain

Type of Data	Examples
Static Data	- Structure of the manufacturing system (i.e. machines, segmentation into machine groups, work areas) - Product information - Setup information - Route information (i.e. information on the process flow)
Dynamic Data	- Lot release information - Lot states - Machine states - Setup states of the machines
Control Data	- Lot Plans - Schedules of different degrees of detail

the aggregation and composition techniques for class diagrams. We use the data model as a starting point to derive important concepts for the domain under discourse. Most of these concepts are covered by the Enterprise Ontology [16] and by the OZONE ontology [13]. However, we can only use the abstract concepts (such as activity, resource, order, product, plan), and we have to add several refinements (sub-concepts) necessary to apply the concepts from [13] and [16] to our domain and the hierarchical production control approach. Most of the necessary extensions are caused by the hierarchical decomposition of the shop-floor and the process flows (called routes).

Based on the data model, we derive the following important basic concepts of the domain.

1. **Activity:** An activity represents a process that can be executed over a time interval. We have to distinguish between controllable activities that can be arranged by a decision-maker, uncontrollable and predictable activities and uncontrollable and unpredictable activities. Important controllable activities are production steps and production operations that are formed by a set of consecutive production steps for one work area. The concept activity and related concepts are shown in UML notation in Figure 2. We include domain-specific concepts like rework activities (cf. [14] for a classification of activities in semiconductor manufacturing).
2. **Resource:** A resource is an entity that enables the execution of activities. According to the OZONE ontology [13] we have to distinguish between single machines and aggregated resources. We consider the special aggregated resources Fab machinery, work area machinery and machine group due to the hierarchical decomposition of the production control problem as described in Section 2. Furthermore, we consider batch and non-batching machines.
3. **Tool:** A tool (also called auxiliary resource) is an entity that supports the execution of activities by resources. Photolithography masks (patterns that contain

the information of an integrated circuit) are important examples of auxiliary resources in semiconductor manufacturing.

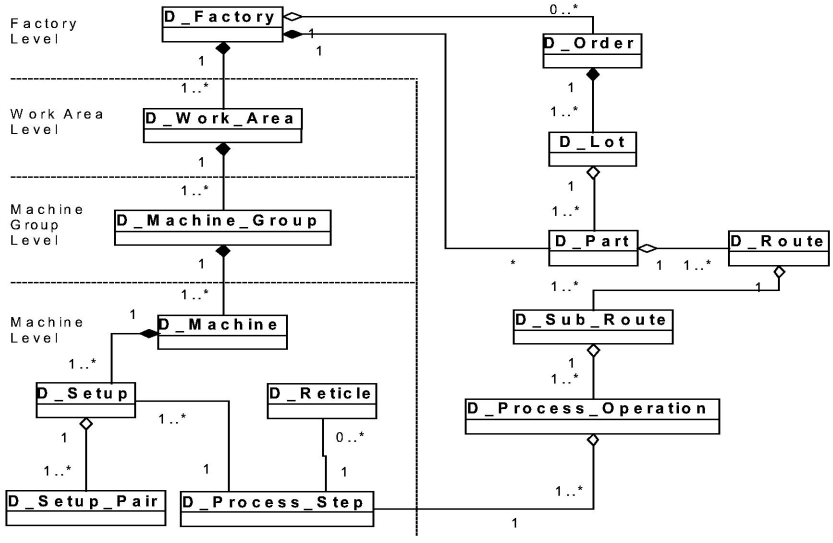


Fig. 1. Data Model for Hierarchically Organized Manufacturing Systems

4. **Order:** An order is a request for a certain number of integrated circuits. An order is characterized by a delivery date, a weight (priority), an order state and the quantity of integrated circuits associated with the order. Each order consists of a certain number of lots. A batch is a temporary collection of lots with the aim to process them at the same time on the same machine.
5. **Product:** A product (also called part) is a piece of goods that meets a set of prescribed requirements. A product is associated with a certain technology. A technology is defined by a sequence of processing steps called route. Usually a product is also associated with a set of auxiliary resources. The route consists of several sub routes that corresponds to the process steps of a certain work area and mask level in the case of the occurrence of reentrant flows.
6. **Event:** An event is a process that starts instantaneously and has no duration. We consider, for example, machine failure events, machine repair events and rework events.
7. **Plan:** A plan is an activity specification with an intended purpose. A plan may include entities that are not released into the manufacturing system yet. A lot plan is a special plan that determines the start and end dates for each production operation of a single lot (released or not).
8. **Schedule:** A schedule is an activity specification with an intended purpose. A schedule only includes entities that are already released into the manufacturing system. In correspondence with the hierarchical production control approach, we consider work area schedules, machine group (work center) schedules and single machine schedules.

Note, that the concepts activity, resource, product, plan, and schedule are highly influenced by the used hierarchical production control approach.

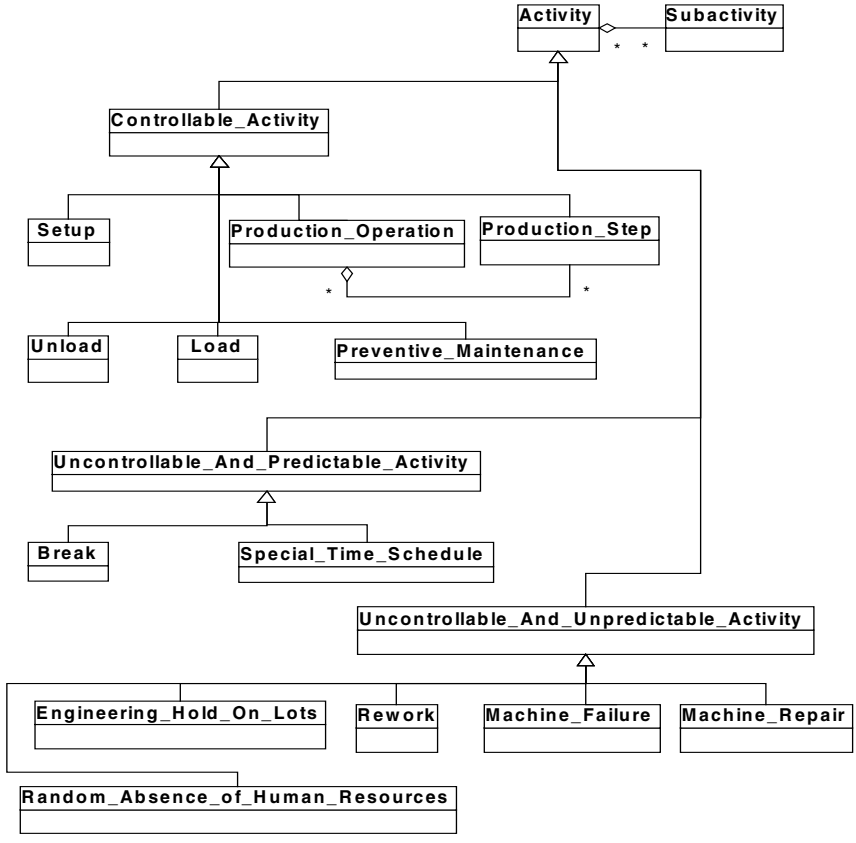


Fig. 2. Concept Activity and Related Sub Concepts of the FABMAS Ontology

In the course of developing the FABMAS ontology we combine techniques described in [15] in order to derive sub-ontologies from a broader ontology (i.e. using the OZONE and Enterprise ontology) with necessary extensions due to the specific semiconductor manufacturing domain.

After deriving important concepts we have to identify predicates. We can distinguish between domain- and task-related predicates. Domain related predicates are facts about a concrete manufacturing system (called basic system), whereas task predicates are closely related to the fulfillment of the goals of the production control system (called control system). In Table 4 we present examples for both domain and task predicates.

Agent activities are important in order to ensure the design goals of the FABMAS multi-agent-system. In most cases we use the agent activities in order to submit instructions from a decision-making agent to its corresponding staff agent. We show

examples of agent activities in Table 5. We use the tool Protégé in order to develop and maintain the proposed ontology.

Table 4. Predicates in the FABMAS Ontology

Predicate	Description
Domain-related	
- CONTAINED_IN_WORKAREA	Determines whether a given machine is contained in a given work area or not.
- PROCESS_STEP_ALLOWED_ON_MACHINE	Determines whether a given process step can be performed on a given machine or not.
Task-related	
- SCHEDULE_VALID	Determines whether a given schedule is valid or not.
- LOT_COMPLETED	Determines whether a given lot is completed or not.
- LOT_LATE	Determines whether a given lot is late with respect to a given segment of the manufacturing system or not.

4 Interactions between the Agents Using the Ontology

We identify certain scenarios for agent interaction. Interaction diagrams are part of efforts to design a coordination language. Here, an appropriate coordination language consists of a set of speech acts, a set of interaction protocols and a suitable ontology.

We consider the request of a work area schedule as an example of using the FABMAS ontology. In this particular case interactions between a lot agent, the work area agent and the work area scheduling agent are required. The lot agent asks the corresponding work area agent for a schedule for its work area. Consequently, the lot agents specify a certain maximum amount of time for determining the schedule (schedules are determined usually in a time-driven manner). The work area agent makes decisions on the scheduling horizon, an appropriate performance measure and a strategy for considering sub problems of the scheduling on a work center level. After this decision-making process the work area agent requests a schedule from the work area scheduling agent. We present the overall scenario in form of an UML sequence diagram in Figure 3.

In Figure 4 we present a more detailed model of the interaction between work area agent and work area scheduling agent. We use SDL (Specification and Description Language) diagrams in order to describe the interactions.

We use agent activities, predicates and concepts of the FABMAS ontology for the above scenario. We start with describing the interaction between lot agent and work

area agent. The lot agent sends a message to the corresponding work area agent with following content

(RECEIVE_SCHEDULE_BY_LOT
:WORK_AREA
:LOT
:SCHEDULING_HORIZON)(NOT(TIME_REACHED :MAX_TIME).

The message content starts with the keyword RECEIVE_SCHEDULE_BY_LOT, an agent activity from the FABMAS ontology. The agent activity is followed by a more detailed description provided by the concepts WORK_AREA, LOT and SCHEDULING_HORIZON from the FABMAS ontology. The agent activity depends on a predicate, in this example, we ask for the maximum scheduling time MAX_TIME.

The interaction between work area agent and work area scheduling agent is more sophisticated (cf. Figure 4 for the complete interaction). In a first step the work area agent initiates the calculation of a schedule by the work area scheduling agent. Therefore, it sends a message with the following content

(START_CALCULATE_SCHEDULE
:WORK_AREA)(NOT(TIME_REACHED:MAX_TIME)).

If the scheduling agent agrees to the request for determining a schedule, the work area agent submits a message with content

(PREPARE_SCHEDULING
:WORK_AREA
:SCHEDULING_HORIZON
:PERFORMANCE_MEASURE
:SCHEDULING_METHOD)(NOT(TIME_REACHED:MAX_TIME)).

Again, PREPARE_SCHEDULING is an agent activity from the FABMAS ontology that is followed by the relevant concept identifiers from our ontology. After obtaining

Table 5. Agent Activities in the FABMAS Ontology

Agent Activity	Description
START_CALCULATE_SCHEDULE	Asks a special scheduling agent to determine a schedule.
START_MONITORING	Asks a special monitoring agent to prepare a certain monitoring report.
SEND_SCHEDULING_RESULTS	Asks a scheduling agent to transfer schedules back to the requesting agent.
PREPARE_SCHEDULING	Asks a scheduling agent to read all the necessary information in order to calculate a schedule.
START_EXCHANGE_INFORMATION_FOR_ABSBH	Asks to initiate an information exchange in course of the area-based shifting bottleneck heuristic (ABSBH).
START_MHR_NEGOTIATION	Asks to initiate the machine hour-rate-based contact net (in case of an invalid schedule).

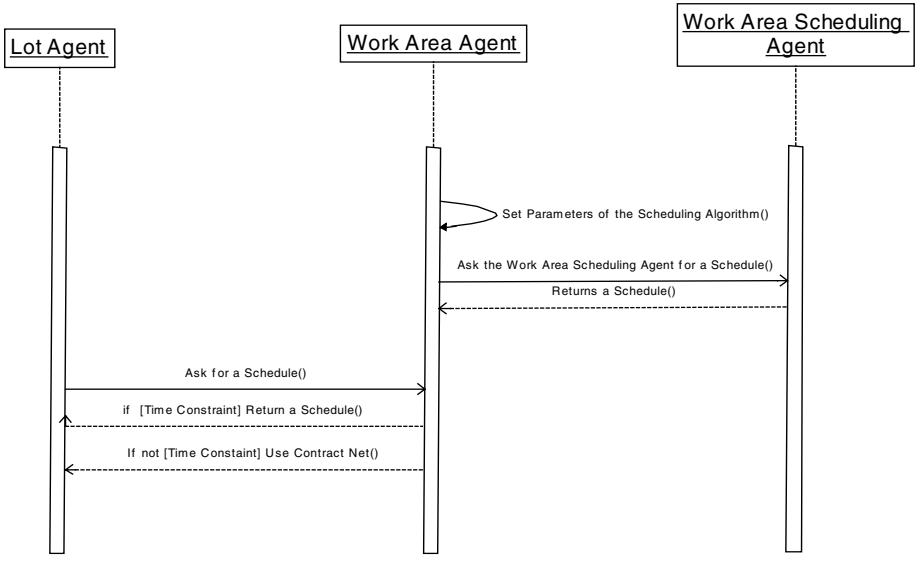


Fig. 3. UML Sequence Diagram for Interaction between Lot, Work Area, and Work Area Scheduling Agent

this message, the work area scheduling agent starts reading all the required information in order to calculate the schedule. Next, the work area agent sends a message with the following content to the work area scheduling agent in order to start the determination of the schedule:

(CALCULATE_SCHEDULE
:WORK_AREA
:SCHEDULING_HORIZON
:PERFORMANCE_MEASURE
:SCHEDULING_METHOD)(NOT(TIME_REACHED:MAX_TIME)).

The content of this message is given by the agent activity CALCULATE_SCHEDULE. In a last step, the work area agent asks for the final schedule by using the content:

(SEND_SCHEDULING_RESULTS
:WORK_AREA
:WORK_AREA_SCHEDULE
:SCHEDULING_HORIZON
:PERFORMANCE_MEASURE
:SCHEDULING_METHOD).

After successful calculation of a schedule, the work area scheduling agent sends the message

(SUCCESSFULLY_DONE:CALCULATE_SCHEDULE)

with predicate type content to the work area agent. In the case of errors during the scheduling determination, other types of content information are necessary (cf. Figure 4).

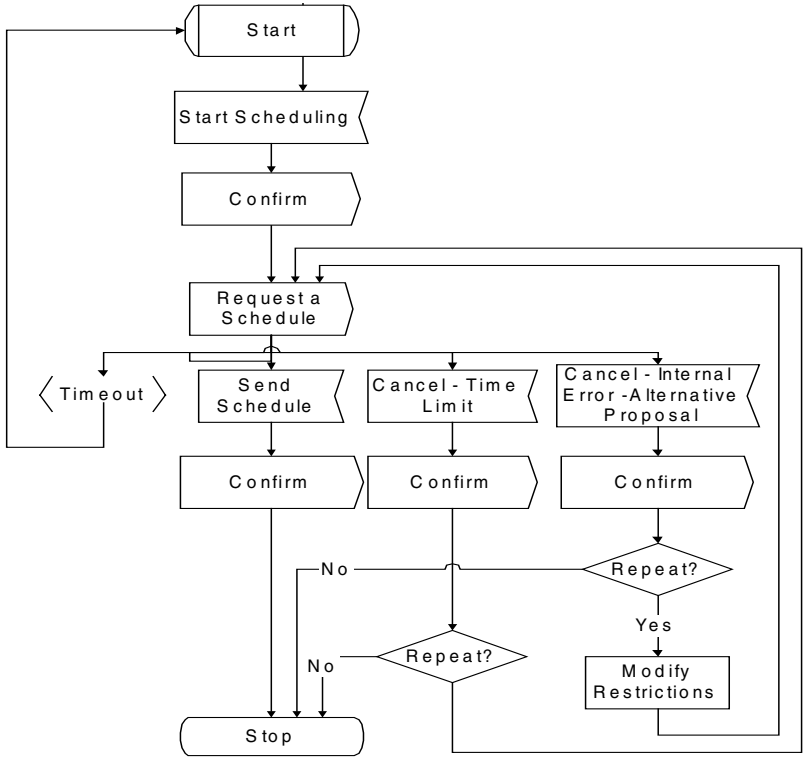


Fig. 4. SDL Diagram for Interaction between Work Area and Work Area Scheduling Agent

5 Conclusions

We present an ontology for production control of semiconductor manufacturing processes. Starting from a brief description of the underlying manufacturing domain and the used hierarchical production control approach, we suggest a data model for the present situation. Using this data model and certain reference ontologies we developed an ontology for the present situation. We describe the use of the ontology in an example.

As an important part of future research we identified the use of the suggested ontology for the construction of a domain-dedicated content language. This language has to be embedded in the FABMAS multi-agent-system. Similar to [6], we do not follow the more generic way of FIPA languages like SL0 or SL1 [1] in order to make the parser for the content language as simple as possible.

Acknowledgement. This research was supported by a grant from the Deutsche Forschungsgemeinschaft (DFG) Priority Research Program 1083 “Intelligent Agents and Realistic Commercial Application Scenarios”.

References

1. Caire, G.: Jade Tutorial: Application-defined Content Languages and Ontologies. TILab S.p.A. (2002)
2. FABMAS Project Web Site: http://www.tu-ilmenau.de/fabmas (2003)
3. FIPA Ontology Specification: http://www.fipa.org/specs/fipa00006/ (1998)
4. Gruber, T. R.: Towards Principles for the Design of Ontologies Used for Knowledge Sharing. Int. J. Hum. Comp. Studies 43 (5/6) (1995) 907–928
5. Habenicht, I., Mönch, L.: A Finite Capacity Beam-Search Algorithm for Production Scheduling of Semiconductor Wafer Fabrication Facilities. In: Yücesan, E., Chen, C.-H., Snowdown, J. L., Charnes, J. M. (eds.): Proceedings of the 2002 Winter Simulation Conference, San Diego, USA (2002) 1406–1413
6. Heikkilä, T., Kollingbaum, M., Valckenaers, P., Bluemink, G.-J.: An Agent Architecture for Manufacturing Control: ManAge, Computers in Industry 46 (2001) 315–331
7. Mesarovic, M. D., Y. Takahara: Abstract System Theory, Lecture Notes in Control and Information Sciences, Springer-Verlag, Berlin, Heidelberg, New York, London, Paris, Tokyo (1989)
8. Mönch, L.: Analyse und Design für ein agentenbasiertes System zur Steuerung von Produktionsprozessen in der Halbleiterindustrie. In: Proceedings Verbundtagung Verteilte Informationssysteme auf der Grundlage von Objekten, Komponenten und Agenten (vertIS 2001) 99–112
9. Mönch, L., Stehli, M., Schulz, R.: An Agent-Based Architecture for Solving Dynamic Resource Allocation Problems. In: Verbraeck, A., Krug, W. (eds.): Proceedings of the 14th European Simulation Symposium. Simulation in Industry, Dresden (2002) 331–337
10. Mönch, L., Stehli, M., Zimmermann, J.: FABMAS – an Agent-Based System for Production Control of Semiconductor Manufacturing Processes. To Appear in Proceedings of the First International Conference on Industrial Application of Holonic and Multi-Agent-Systems (HoloMAS 2003). Prague, Czech Republic (2003)
11. Obitko, M., Mařík, V.: Ontologies for Multi-Agent-Systems in Manufacturing Domain. In: Proceedings of the 13th International Workshop on Database and Expert Systems Applications (2002)
12. Schneeweiß, C.: Hierarchies in Distributed Decision-Making. Springer-Verlag (1999)
13. Smith, S., Becker, M.: An Ontology for Constructing Scheduling Systems. In: Working Notes of the 1997 Symposium on Ontological Engineering, AAAI Press (1997)
14. Srivatsan, N., Bai, S. X., Gershwin, S. B.: Hierarchical Real-Time Integrated Scheduling of a Semiconductor Fabrication Facility. Control and Dynamic Systems (61) (1994) 197–241
15. Wouters, C., Dillon, T., Rahayu, W., Chang, E.: A Practical Walkthrough of the Ontology Derivation Rules. In: R. Cichetti et al. (Eds.): DEXA 2002, LNCS 2453, 259–268, Springer-Verlag, Berlin Heidelberg (2002)
16. Uschold, M., King, M., Moralee, S., Zorgios, Y.: The Enterprise Ontology, http://www.aiai..ed.ac.uk/~enterprise/enterprise/ontology.html#papers, (1997)
17. Van Brussel, H., Wyns, J., Valckenaers, P., Bongaerts, L., Peeters, P.: Reference Architecture for Holonic Manufacturing Systems: PROSA. Computers in Industry, Special Issue on Intelligent Manufacturing Systems 37(3) (1998) 225–276