S^{2}CMAS: An Agent-Based System for Planning and Control in Semiconductor Supply Chains

Raphael Herding and Lars Mönch(✉)

Chair of Enterprise-Wide Software Systems, University of Hagen,
Universitätsstraße 1, 58097 Hagen, Germany
{Raphael.Herding,Lars.Moench}@fernuni-hagen.de

Abstract. In this paper, we describe the design and the implementation of an agent-based system that supports planning and control activities in semiconductor supply chains. The proposed system extends the FABMAS prototype for production control of single wafer fabs by additional enterprise-wide planning-related decision-making agents and staff agents. Web services are used to implement parts of the new planning functionality. Results of simulation experiments with the proposed multi-agent system prototype are presented that indicate that the proposed approach is feasible.

Keywords: Semiconductor supply chains · Agent-based decision support · Design of multi-agent systems · Web services · Simulation-based performance assessment

1 Introduction

Semiconductor manufacturing is characterized by a set of complex manufacturing processes to produce integrated circuits, i.e. chips. The manufacturing process starts with thin silicon discs, so called wafers. Several hundred chips can be produced on each single wafer by fabricating the chips layer by layer in a wafer fab. Then electrical tests identify the dies that are likely to fail when packaged in a probe/sort facility. Probed wafers are sent to assembly facilities where dies that fulfill the requirements are put into a package. In a last step, the assembled dies are sent to a test facility where they are tested to ensure that only high-quality products are delivered to customers. Wafer fab and sort operations are subsumed under frontend operations, whereas assembly and test operations are combined into backend operations. The manufacturing process in the semiconductor industry is very complex because of reentrant process flows on extremely expensive machines, long cycle times, and multiple sources of uncertainty. Capacity expansions are time-consuming and very expensive. The related decisions are based on forecasts that are rarely accurate since the demand is highly volatile [1, 18].

Supply chain management (SCM) issues have become more and more important in the last decade for the semiconductor domain. This is because of the fact that front-end operations are often performed by highly industrialized countries, while backend operations are typically carried out in countries where the labor rates are cheaper.

M. Klusch et al. (Eds.): MATES 2016, LNAI 9872, pp. 115–130, 2016.
DOI: 10.1007/978-3-319-45889-2_9

Semiconductor supply chains are an extreme field for SCM solutions from an algorithmic and also from a software point of view [1]. At the same time, the decision support for semiconductor supply chains is often based on packaged Advanced Planning and Scheduling (APS) software [11, 14, 21] that is not tailored to the specific needs of semiconductor supply chains.

While some multi-agent systems (MAS) are proposed for operational decisions in single wafer fabs [15, 16, 29] to the best of our knowledge, this is not the case for semiconductor supply chains. In the present paper, we present design and implementation details for the S^{2}CMAS prototype, a MAS that is hybridized with web services to provide planning and control functionality for semiconductor supply chains.

The paper is organized as follows. In the next section, we describe the problem. Therefore, we identify requirements for an agent-based decision support in enterprise-wide semiconductor supply chains. In addition, related literature is discussed. The design of the S^{2}CMAS prototype is specified in Subsect. 3. The results of simulation experiments with the prototype are presented in Subsect. 4. Conclusions and future research directions are provided in Sect. 5.

2 Problem Setting

2.1 Requirements for Agent-Based Planning and Control

The planning functionality and control functionality for semiconductor supply chains include demand and inventory planning, network design, long and mid-term capacity planning, master planning, order release planning, i.e. operational production planning, and various production control activities such as scheduling and dispatching. Demand and inventory planning, network design, capacity planning and master planning are performed on the network level, while order release planning and scheduling and dispatching is often carried out for single frontend and backend facilities. A distributed hierarchical decision making process (cf. [24] for details) is implemented in semiconductor supply chains [18] where two entities are in a hierarchical relationship if they exhibit some asymmetric relationship with respect to their decision rights, their information status, or simply the point of time where decisions are made [24].

It is well-known that monolithic packaged software systems for planning and control purposes have some limitations in semiconductor supply chains (cf. [14, 21, 23]) since it is challenging to deal with the demand uncertainty, the long cycle times, and the reentrant flows. Therefore, companies tend to offer more specialized functionality by extending commercial software packages or using fully home-grown solutions. In this situation, the packaged software systems often provide the data for the tailored systems. However, since supply chains are distributed and contain by nature decision-makers with a certain level of autonomy, it seems that planning and control functionality offered by software agents provides some advantage in this situation.

Next, we summarize requirements that have to be fulfilled by a planning and control system for semiconductor supply chains:

1. The total functionality of the system must span the entire range of planning and control functions required to manage a semiconductor supply chain ranging from

long-term demand and inventory planning for the entire network to short-term detailed scheduling for single wafer fabs. Therefore, the distributed hierarchical structure of semiconductor supply chains has to be taken into account by the system architecture. This means that a decentralized decision-making has to be allowed by the planning and control system. This includes especially iterative feedforward and feedback cycles to implement communication or even negotiation processes among the different decision-making entities.

2. The system has to be able to communicate with other application systems and human decision makers in the semiconductor supply chain to foster an information exchange. Existing communication standards have to be respected if appropriate.
3. It is required that the planning and control system is scalable, i.e., increasing the size of the base system and the base process does not lead to a loss in performance of the provided planning and control algorithms.
4. The frequent changes in the base system and process of semiconductor supply chains have to be reflected by the system architecture, i.e., it is desirable that certain functionality of the overall planning and control system can be replaced during run time of the system. Therefore, a clear separation between the planning and control algorithms and the planning and control system is required.
5. The planning and control system has to allow for a situation-dependent parameterization of the provided planning and control algorithms.
6. Since the system must be able to support planning decisions, it is required that appropriate what-if analysis functionality is provided. That means especially that the system is able to derive meaningful variants of given base scenarios to analyze and understand the consequences of future decisions.
7. The system must be capable to support assessing the performance of the planning and control algorithms using simulation. Simulation can also be used as a tool that supports decisions or can be used as an important ingredient of a what-if analysis.
8. The system must be able to integrate functionality that is provided by other application systems in the semiconductor supply chain since it cannot be expected that the proposed system provides the entire production planning and control functionality.
9. There is often more than one service candidate to provide a specific functionality. In this situation, an appropriate selection of service candidates has to be supported by the proposed system.
10. The system architecture must be open to adopt new trends in manufacturing such as cloud computing [8] or in-memory computing [7].

According to Schneeweiss [24] it is possible to implement distributed hierarchical decision-making systems by means of MAS if the corresponding software agents are equipped with optimization and advanced coordination capabilities. At the same time, agents allow for a meaningful, rich communication. Since MAS are distributed systems, the scalability of the system is ensured. Therefore, we are interested in providing agent-based decision support for semiconductor supply chains to fulfill the requirements 1–4.

The concept of decision-making and staff agents is provided by the Product Resource Order Staff Architecture (PROSA) reference architecture [27] in holonic manufacturing. Staff agents support decision-making agents in course of solving their decision problems. They encapsulate algorithms for decision-making. PROSA is extended towards generic application domains by differentiating between resources, tasks, and task types in [26]. We argue that a separation between decision-making and staff agents is useful to fulfill the requirements 5–7. A separation of planning and control algorithms from the planning and control system is ensured by differentiating between decision-making and staff agents. Staff agents are able to provide situation-dependent parameter settings in the various planning and control algorithms, i.e. requirement 5 is supported. Scenario agents as specific staff agents fulfill requirement 6. Discrete-event simulation can be embedded into staff agents to meet the requirement of simulation-based support of performance assessment and decision-making capabilities.

The remaining three requirements are supported by hybridizing the MAS with web services. Planning and control functionality provided by legacy systems can be encapsulated into web services. The operations of these web services can be used by the software agents, i.e., the web services can be discovered and engaged by software agents. Therefore, this allows us to meet requirement 8. Techniques for web service selection can be incorporated into staff agents. This results in supporting requirement 9. Finally, requirement 10 is supported by web services since web services are considered as one of the key enabling technologies for cloud manufacturing [7]. Moreover, web services support to a certain extent requirement 2 since they are based on human readable and machine readable, i.e., XML-based communication protocols.

2.2 Related Work

We discuss related work with respect to MAS in SCM, methods to hybridize service-oriented architecture (SOA) approaches with MAS and known MAS applications in semiconductor manufacturing. Agent-based approaches for SCM are discussed quite often in the literature. Requirements for agent-based systems to provide SCM functionality are discussed in [5]. We refer to [13] for a related survey.

There are several attempts described in the literature to equip MAS with SOA techniques and vice versa. Huhns [9, 10] argues that agents, on the one hand, have to applied in SOA-based information systems to support service composition and selection, to contribute the rich communication abilities, and to allow for a proactive behavior of the resulting hybrid systems. On the other hand, SOA solutions offer operability in a standardized way, and they are widely accepted in the industry. A similar discussion is presented in [28].

The ALIVE architecture described in [3] differentiates between a coordination layer that is based on software agents and a service layer that is formed by web services. The web services are invoked by the software agents of the coordination layer. The differentiation between agents and services in ALIVE is somewhat similar to the differentiation of decision-making and staff agents in the PROSA architecture, i.e., the staff agents from PROSA are replaced by web services. Decision support systems based on software agents that invoke services to build and run decision models are proposed in [4]. The proposed architecture is applied to solve a collaborative planning, forecasting and replenishment problem in supply chains. In a certain sense, this proposal is similar to replacing certain services provided by a staff agent in PROSA by web services.

An agent-based approach for supply chain collaboration is discussed in [12]. Web services are enriched by coordinating software agents to ensure autonomy and proactiveness of the decision-making entities in the supply chain. Combining agents with web services for decision-making in supply chains is also proposed in [22]. However, rather simple supply chains compared to semiconductor supply chains are considered in these two papers. The Supply Chain Optimization and Protocol Environment (SCOPE) [19] allows for a rapid prototyping of supply chains. The nodes of a supply chain are represented by simple agents that are able to make production planning decisions based on linear programming models.

It is expected that industrial agent-based systems for automation will incorporate SOA concepts since web services can be implemented directly on devises [2]. The agents can be integrated with the devises or act as a service orchestrator. A framework for developing service-oriented agent-based manufacturing systems is proposed in [6]. It equips the ANEMONA approach for developing holonic manufacturing systems with service-based features that are required to apply ANEMONA for the SCM domain. However, in contrast to [6], we believe that the decision problems should be the starting point to design MAS applications for complex supply chains.

MAS approaches applied to the semiconductor manufacturing domain are rarely discussed in the literature. We are only aware of the FABMAS system prototype [15, 16] and a MAS for scheduling with hard real-time restrictions in wafer fabs described in [29]. FABMAS is a hierarchically organized MAS that provides scheduling support for single wafer fabs. The FABMAS design is based on PROSA. The system is implemented using the ManufAg framework [17]. SCOPE is applied in [20] to model fairly simple semiconductor supply chains. Simulation models based on system dynamics are used to study the dynamic behavior of the supply chain. However, the strengths of agent-based approaches such as rich communication capabilities or distributed computing abilities are not fully used in this paper. Overall, to the best of our knowledge, agent-based systems for semiconductor supply chains are not described in the literature so far. Based on the discussion in this subsection, it seems reasonable to develop hybrid systems that involve software agents and web services at the same time.

3 Design and Implementation of the S^{2}CMAS Prototype

3.1 Identifying Appropriate Agents

We start by identifying appropriate software agents. This step should be based on the analysis of the decision problems to be solved. Because of requirement 1, it makes sense to extend the FABMAS system towards the new functionality needed to carry out planning and control tasks in semiconductor supply chains. This means especially that the agents provided by FABMAS will be included in S^{2}CMAS too. The most important decision-making and staff agent types summarized in Table 1 are considered in the S^{2}CMAS prototype. The abbreviation DM refers to decision-making agents while S indicates that a staff agent is used.

Table 1. Main functionality of the members of the FABMAS agency

Agent	Type	Description
Lot agent	DM	Represents a single lot
Product agent	DM	Encapsulates the product knowledge
		Decision-making for selecting alternative machines in the case of machine breakdowns
Fab agent	DM	Coordinates the lot planning agent
		Coordinates the work area agents
		Decision-making for lot-based decomposition approaches
Lot planning agent	S	Prepares to run a specific lot planning algorithm
		Runs the algorithm
		Provides lot plans
Work area agent	DM	Coordinates the work of the corresponding work area scheduling agent
		Decision-making in form of choosing appropriate shifting bottleneck heuristic ingredients
Work area scheduling agent	S	Prepares to run the shifting bottleneck heuristic
		Runs the heuristic
		Provides scheduling information
Work center agent	DM	Implements the work area schedules
		Serves as mediator for contract net-based allocation algo-Rythms

The agents in Table 1 exploit the fact that lots, a collection of wafers, are the moving entities in wafer fabs, and that a single wafer fab can be decomposed into several work areas. Each work area consists of different work centers. A single work center is formed by machines that provide the same functionality. The fab agent is responsible for decomposing the overall scheduling problem for a single wafer fab into a series of scheduling problems for work areas. Start and completion dates for the different work areas are assigned to each single lot using lot planning algorithms. The scheduling problem for a single work area is solved by the shifting bottleneck heuristic. Iterative exchange processes to compute modified start and completion dates based on the work area schedules are performed (cf. [16] for details of the algorithms).

Based on the additional production planning and control functionality described in Subsect. 2.1, new decision-making agent types are identified for the S^2CMAS system to support planning and control activities in the supply chain. We take into account the distributed hierarchical structure of semiconductor supply chains when identifying the agents. This approach again supports the fulfillment of requirement 1. The identified decision-making agent types are summarized in Table 2.

Note that several refinements and extensions of the agency in Table 2 are possible. For instance, a decision-making agent for inventory management might be added. We see from Table 2 that most of the identified agents are responsible for network-wide tasks whereas the agents in Table 1 only provide functionality to control a single wafer fab.

Table 2. Main functionality of the additional DM agents of the S^{2}CMAS agency

Agent	Description
Demand planning agent	Coordinates the forecasting agent
	Coordinates the capacity planning agent
Capacity planning agent	Coordinates the long-term network-wide planning agent
	Coordinates the master planning agent
	Interacts with scenario agents
Master planning agent	Coordinates the different order release agents
	Coordinates the mid-term network-wide planning agents
	Interacts with scenario agents
Order release agent	Coordinates the fab planning agent
	Coordinates with the fab agent
Available to Promise (ATP) agent	Coordinates the ATP planning agent
	Interacts with the master planning agent
Order agent	Represents a single order (based on a customer request or on forecast)

Next, we summarize various staff agents in Table 3 that support the agents from Table 2 in the course of their decision-making. Staff agents encapsulate decision rules and support the decision-making agents to perform actions. We see from Table 3 that several staff agents exist that provide planning functionality. Moreover, the scenario agents and the simulation agents might provide support for several decision-making agents that have to provide planning functionality. The simulation agent offers discrete-event simulation functionality. Services encapsulate a concrete functionality that is stateless. They might offer a concrete planning or simulation functionality.

Table 3. Main functionality of the additional staff agents of the S^{2}CMAS agency

Agent	Description
Forecasting agent	Prepares to run a specific forecast algorithm
	Runs the algorithm
	Aggregates/disaggregates the forecasts
	Provides forecast information
Long-term network-wide planning agent	Prepares to run a specific planning algorithm
	Runs the algorithm
	Provides plans for the network
Mid-term network-wide planning agents	Prepares to run the planning algorithm
	Runs the planning algorithm
	Provides master plans
Fab planning agent	Prepares to run a specific production planning algorithm
	Runs the algorithm for a single wafer fab or a set of wafer fabs
	Runs the algorithm
	Provides release plans for single wafer fabs

(*Continued*)

Table 3. (*Continued*)

Agent	Description
Scenario agent	Generates planning scenarios for various planning-related DM agents based on forecast evolution and capacity pattern
	Provides the scenarios
Simulation agent	Simulates plans to determine expected values of the performance measures
	Supports what-if functionality
ATP planning agent	Prepares algorithms for order acceptance/selection, due date assignment, and order scheduling
	Runs the algorithm
	Proves ATP-related information to master planning and order management
Order management agent	Provides order processing functionality
	Provides order status-related information

3.2 Providing Planning Functionality by Web Services

The different staff agent types identified in Subsect. 3.1 offer the following generic functionality:

1. Select an appropriate service that provides the requested functionality.
2. Prepare to use the specific functionality by collecting the required data.
3. Based on the data from Step 1, a situation-dependent parameterization of the requested functionality, for instance a specific planning algorithm, is performed.
4. Use the functionality, for instance, run the planning algorithms. This might include terminating the functionality based on specific events, for example, by reaching a given maximum computing time.
5. Inform the corresponding decision-making agents about the results of the previous steps.

Since it is unlikely that the entire planning and control functionality will be developed from scratch, decision-making and staff agents might use planning and control functionality from legacy systems. Therefore, it is desirable that the functionality is offered by web services since many packaged software systems are equipped with web services. Composite services can be used by orchestrating existing services. Step1 of the generic functionality ensures that an appropriate service candidate is chosen for each task of a service composition. This requires the solution of appropriate optimization problems.

We differentiate three different ways how services can be used by agents in the S^{2}CMAS prototype. The first approach is based on the idea that a web service is directly invoked by a decision-making agent. Such a setting is reasonable if only a single service candidate exists and if most of the generic functionality for staff agents is not required. This means especially that a situation-specific parameterization of the functionality does not offer much value. The second option uses a staff agent between

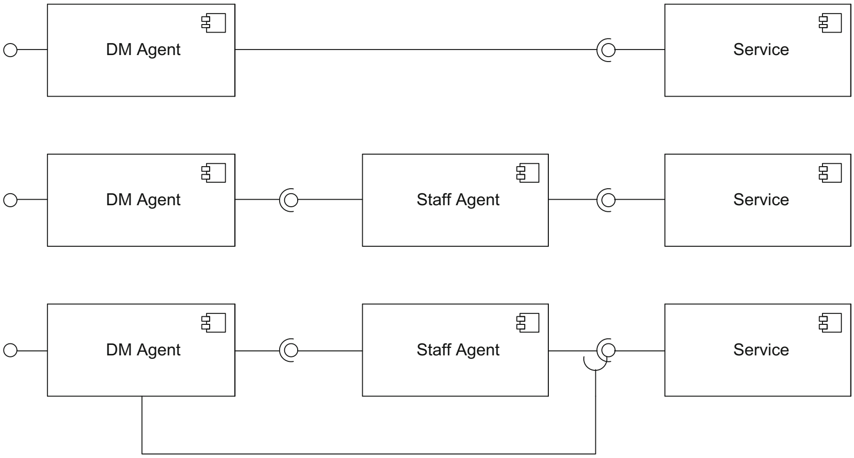

Fig. 1. Usage of services by agents in the S^{2}CMAS prototype

the web service and the decision-making agent. This option is favorable if the full staff agent functionality is required. In a certain sense, staff agents serve as orchestrators. The third option, however, hybridizes the first and the second option. Here, services can be invoked by decision-making agents and by staff agents depending on the problem to be solved. The three principle approaches are shown in Fig. 1.

3.3 Implementation Issues

The S^{2}CMAS prototype is implemented in the C# programming language by extending the FABMAS system [15, 16] that is based on the ManufAg framework [17]. ManufAG allows for implementing distributed hierarchically organized MAS. The web services are stateless. They are coded in the C# programming language. The communication between the agents of the MAS and the web services is based on the HTTP protocol.

The simulation agents are based on simulation services provided by the commercial simulation framework AutoSched AP 9.3 that is coded in the C ++ programming language. The simulation agents build and parameterize AutoSched AP simulation models in a problem- and situation-specific manner. Moreover, a preliminary analysis of the simulation results is performed by these agents. Appropriate demand signals are generated by a demand generator that is implemented in the C# programming language.

Discrete-event simulation is used to assess the S^{2}CMAS prototype. The center point of the proposed architecture is a blackboard-type data layer coded in the C ++ programming language in the memory of the simulation computer that is between the S^{2}CMAS prototype and a simulation model of the base system of the semiconductor supply chain. It contains all the relevant business objects such as lots, machines, and products with corresponding routes. These objects are updated whenever status changes occur in the simulation using the notification mechanism of the simulation engine AutoSched AP. The overall architecture is shown in Fig. 2.

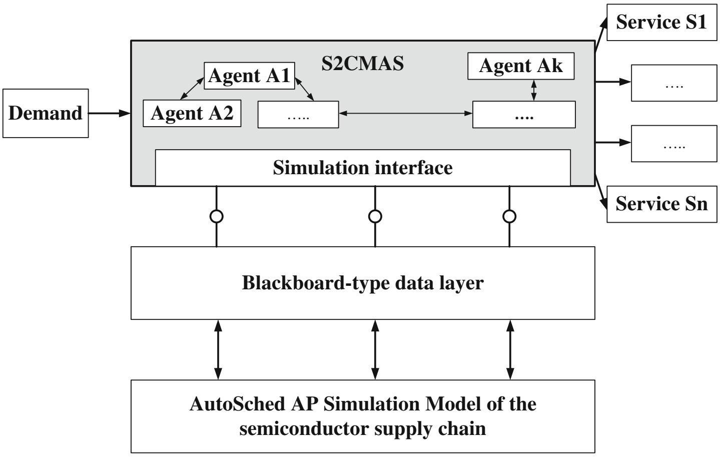

Fig. 2. Performance assessment architecture

4 Simulation Experiments with the S^{2}CMAS Prototype

4.1 Provided Planning Functionality

A finite planning horizon of length T divided into discrete periods of equal length is given. Deterministic demand information is available for the planning horizon. The objective is to determine the amount of each product to release into the wafer fabs in each period so as to minimize the costs caused by these releases. Therefore, we use backward termination to determine the release quantities based on demand D_{gt} for product g in period t. For the sake of simplicity, we assume that D_{gt} is measured in number of lots. We set the release period r_j for each lot j that belongs to D_{gt} by

$$r_j := \left\lfloor \max\left(t - FF \sum_{k=1}^{n_j} p_{jk}, 0 \right) \right\rfloor, \tag{1}$$

where p_{jk} is the processing time of process step k of lot j and n_j is the number of process steps for lot j. The quantity $FF \geq 1$ is the flow factor, i.e. the ratio of the amount of time a lot spends in a wafer fab and the raw processing time $\sum_{k=1}^{n_j} p_{jk}$. Of course, we assume that an amount of $(FF - 1)p_{jk}$ is associated with the waiting time for process step k of lot j. All lots to be released in the current period t are distributed uniformly over this period. The resulting detailed release date for lot j is s_j. The obtained release schedule is implemented until the next backward termination is computed along the timeline in a rolling horizon setting.

The quantity FF has to be chosen in Eq. (1). Since the cycle times of the lots increase in a nonlinear manner with increasing resource utilization that is a result of the release plan, we propose to use discrete-event simulation to find appropriate FF values. We consider the future demand to come up with a FF value for this situation.

The demand leads to a certain bottleneck utilization that can be used to determine a FF value FF_1. In addition, realized FF values from already executed release plans are taken into account to compute a second FF value FF_2. Several FF values from an equidistant grid of the interval $[\min(FF_1, FF_2), \max(FF_1, FF_2)]$ are used to determine release plans. These plans are then simulated using FIFO dispatching of the lots. The *FF* values associated with the release plane are calculated. The FF value $FF^* \in [\min(FF_1, FF_2), \max(FF_1, FF_2)]$ leading to the smallest absolute difference between the measured FF value and the FF value used for the backward termination is selected. Next, the lots to be released in the current period are assigned to individual wafer fabs by applying simple load balancing techniques.

The release plans are used to compute start and completion dates for the lots with respect to each single work area. This information is necessary to determine detailed schedules based on the distributed shifting bottleneck heuristic (DSBH).

The order release planning functionality is implemented in the S^2CMAS prototype by adding an order release agent. This decision-making agent interacts with a single Fab planning agent. This agent is responsible to run the backward termination algorithm that is implemented by means of a web service. In order to parameterize the backward termination algorithm correctly, several simulation runs are necessary that are provided by a simulation agent. Therefore, the second option for invoking services by agents is used in this planning scenario. The final release plan is submitted to the fab agent of a single wafer fab that is responsible to determine lot plans that are a necessary ingredient to compute detailed schedules for the lots in each single wafer fab using the DSBH approach. The described situation is shown in the UML sequence diagram in Fig. 3.

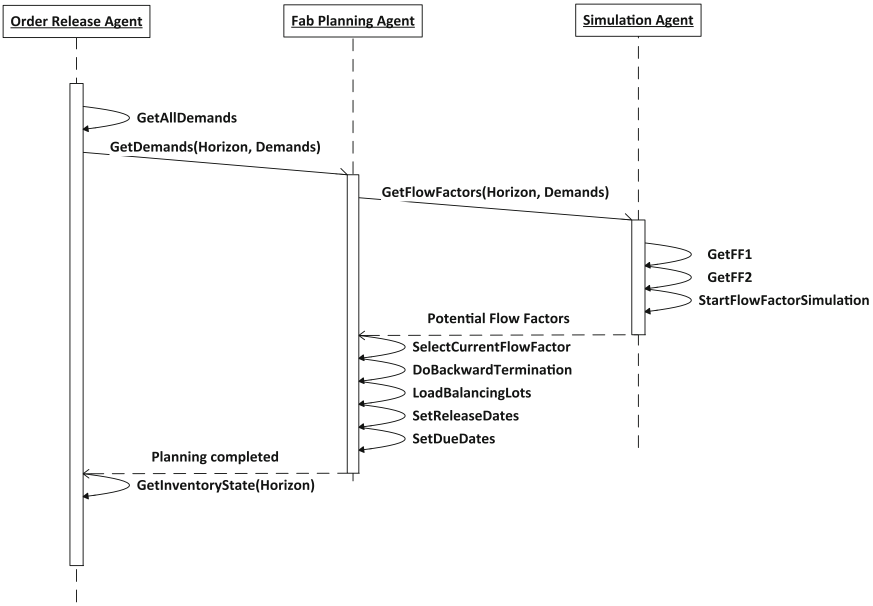

Fig. 3. Interaction of the different agents in the order release planning scenario

4.2 Simulation Environment and Design of Experiments

A small-sized enterprise-wide semiconductor supply chain that consists of two identical wafer fabs is used within the simulation experiments for applying the order release functionality and detailed scheduling in a rolling horizon manner. Each of them contains three work areas. A single work area consists of three work centers, among them one with batch processing machines, i.e., several lots are processed at the same time in a batch on a single machine, and another one with sequence-dependent setup times. The first work area of each wafer fab is visited two times. Each wafer fab has 15 machines and two products with 24 process steps.

We expect that the performance of the order release mechanism depends on the demand setting. Therefore, we generate normally distributed demand that leads to 70 % and 96 % planned bottleneck utilization in the two wafer fabs. In addition, a coefficient of variation of 0.1 and 0.25 is considered to generate the demand. The first setting refers to low variability demand (VL) and the latter to demand with high variability (VH). In addition, we consider two simple approaches to assign release quantities to a specific wafer fab. The first approach is load-based, i.e., the wafer fab with the smallest work in progress (WIP) is assigned to release the lots associated with a certain product and period. The second approach randomly assigns with equal probability one of the two fabs to the lots. The products and the periods are taken into account in a given sequence for both approaches.

Simulation runs are performed for 200 days. The length of a single period is two days. The release plan is revised after a single period, i.e., only the first period is implemented. A planning horizon of $T = 6$ periods is considered. The DSBH is used every two hours with a scheduling horizon of three hours. The scheduling subproblems that are a result of the SBH approach are solved using list scheduling based on the Apparent Tardiness Cost dispatching rule. The lot planning algorithm offered by the lot planning agent is applied twice per shift. Here, a fairly simple forward termination algorithm is used to determine the start and completion dates for the lots with respect to each single work area. A due date d_j is assigned to each released lot based on

$$d_j := s_j + z_j \cdot FF^* \sum_{k=1}^{n_j} p_{jk}, \tag{2}$$

where $z_j = 0.45$ if lot j belongs to the first product and $z_j = 2.35$ if j is a lot of the second product. We use this due date setting scheme to mimic the situation that the processing of lots of the first product is urgent whereas lots of the second product have a large slack. The lot weights are selected as $w_j = 10$ for the first product, whereas the setting $w_j = 1$ is used for the second product. We are interested in assessing the profit obtained over the simulation horizon. The profit is the difference of revenue and WIP, backlog, and inventory holding costs. The revenue value per lot is 180, while the unit backlog, WIP, and inventory costs per period are 50, 35, and 15, respectively. The computational experiments are carried out on a PC with 32 GB of RAM and an Intel Xenon E5-2620 CPU with 24 virtual cores where each core has a CPU frequency of 2.0 GHz.

4.3 Results of Computational Experiments

The simulation results are shown in Table 4. Here, we denote by BN the planned bottleneck utilization, whereas LB is used to indicate the load balancing approach used in the experiments. The results are presented as follows. Instead of comparing all simulation cases individually, the cases were grouped according to factor level values. For example, the results for VL and $BN = 70\,\%$ are averaged over all runs where demand is generated with $CV = 0.1$ that leads to a bottleneck utilization of 70 % while all the other factors have been varied at their different levels. We show in each column first the value obtained by FIFO dispatching, while the corresponding value obtained by the DSBH scheme is in the cell below. For instance, for VL and $BN = 70\,\%$ we have a profit value of 343540 for FIFO and a profit value of 358500 for DSBH. Superior profit values are marked in bold.

We see from Table 4 that using the order release planning approach together with the DSBH-based scheduling approach is beneficial, i.e., the scheduling approach leads to larger profit values compared to FIFO-based dispatching. The positive effects are larger in case of low variability demand and a highly utilized planned bottleneck. The two load balancing schemes do not lead to significantly different profit values. This

Table 4. Simulation results

	Revenue	WIP	Backlog	Inventory	Profit
	FIFO/SBH	FIFO/SBH	FIFO/SBH	FIFO/SBH	FIFO/SBH
VL					
BN					
70 %	474040	103985	11500	15015	343540
	484040	104055	9500	11985	**358500**
97 %	820380	212310	50400	20820	536850
	852020	217210	46150	16145	**572515**
LB					
L	631440	159145	31300	17595	423400
	667940	161805	28350	13720	**464065**
R	631980	157150	30600	18240	425990
	668120	159460	27300	14410	**466950**
VH					
BN					
70 %	461340	102970	11350	14745	332275
	466556	103215	8750	10544	**344047**
97 %	794520	208740	47200	20370	518210
	826160	214340	46900	18680	**546240**
LB					
L	627480	156135	28850	17310	425185
	646088	158690	27000	14402	**445996**
R	628380	155575	29700	17805	425300
	646628	158865	28650	14822	**444291**

result is expected since the lots are randomly released into one of the two wafer fabs with equal probability and the two fabs are identical.

5 Conclusions and Future Research

In this paper, we discussed an agent-based system prototype that supports planning and control activities in semiconductor supply chains. Appropriate agents are identified. We designed the interaction of the various agents with web services that implement the planning and control functionality. The results of experiments with the S^{2}CMAS prototype using a simulation model of a scaled-down semiconductor supply chain were presented and discussed.

There are several directions for future research. First of all, the functionality of the S^{2}CMAS prototype has to be extended. We are interested in adding functionality related to master planning, to the order management process and related to ATP. More simulation experiments with larger simulation models of semiconductor supply chains have to be performed. Standardized communication is an important ingredient for an agent-based decision support. Therefore, we are interested in designing an appropriate ontology based on the ISA 95 standard (cf. Scholten [25]).

Acknowledgement. The authors would like to thank the attendees of the Dagstuhl Seminar 16062 “Modeling and Analysis of Semiconductor Supply Chains”, especially Irvan M. Ovacik, Intel, and Kenneth Fordyce, Arkieva, for interesting discussions about modeling issues and planning software for semiconductor supply chains.

References

1. Chien, C.-F., Dauzère-Pérès, S., Ehm, H., Fowler, J.W., Jiang, Z., Krishnaswamy, S., Mönch, L., Uzsoy, R.: Modeling and analysis of semiconductor manufacturing in a shrinking world: challenges and successes. Eur. J. Ind. Eng. **5**(3), 254–271 (2011)
2. Colombo, A.W., Karnouskos, S., Mendes, J.M., Leitão, P.: Industrial agents in the era of service-oriented architectures and cloud-based industrial infrastructures. In: Leitão, P., Karnouskos, S. (eds.) Industrial Agents: Emerging Applications of Software Agents in Industry, Chap. 4, pp. 67–87. Elsevier, Amsterdam (2015)
3. Dignum, F., Dignum, V., Padget, J.A., Vázquez-Salceda, J.: Organizing web services to develop dynamic, flexible, distributed systems. In: Proceedings of the Eleventh International Conference on Information Integration and Web-based Applications and Services (iiWAS 2009), pp. 225–234 (2009)
4. Dong, C.-S.J., Srinivasan, A.: Agent-enabled service-oriented decision support systems. Decis. Support Syst. **55**, 364–373 (2013)
5. Fox, M.S., Barbuceanu, M., Teigen, R.: Agent-oriented supply-chain management. Int. J. Flex. Manuf. Syst. **12**, 165–188 (2000)
6. Giret, A., Botti, V.: ANEMONA-S + Thomas: a framework for developing service-oriented intelligent manufacturing systems. In: Borangiu, T., Thomas, A., Trentesaux, D. (eds.) Service Orientation in Holonic and Multi-agent Manufacturing, pp. 61–70. Springer, Heidelberg (2015)

7. Hahn, G.J., Packowski, J.: A perspective on applications of in-memory analytics in supply chain management. Decis. Support Syst. **76**, 45–52 (2015)
8. He, W., Xu, L.: A state-of-the-art survey of cloud manufacturing. Int. J. Comput. Integr. Manuf. **28**(3), 239–250 (2015)
9. Huhns, M.N.: Software agents: the future of web services. In: Kowalczyk, R., Müller, J.P., Tianfield, H., Unland, R. (eds.) NODe-WS 2002. LNCS (LNAI), vol. 2592, pp. 1–18. Springer, Heidelberg (2003)
10. Huhns, M.N.: Services must become more agent-like. Wirtschaftsinformatik **50**, 74–75 (2008)
11. Kallrath, J., Maindl, T.I.: Planning in semiconductor manufacturing. In: Kallrath, J., Maindl, T.I. (eds.) Real Optimization with SAP APO, Chap. 5, pp. 105–117. Springer, Heidelberg (2006)
12. Kwon, O., Im, G.P., Lee, K.C.: An agent-based web service approach for supply chain collaboration. Scientia Iranica1 Trans. E Ind. Eng. **18**(6), 1545–1552 (2011)
13. Lee, J.-H., Kim, C.-O.: Multi-agent systems applications in manufacturing systems and supply chain management: a review paper. Int. J. Prod. Res. **46**(1), 233–265 (2008)
14. Lin, C.-H., Hwang, S.-L., Wang, M.-Y. E.: The mythical advanced planning systems in complex manufacturing environment. In: Proceedings of the 12th IFAC Conference on Information Control Problems in Manufacturing, pp. 691–696 (2006)
15. Mönch, L., Stehli, M., Zimmermann, J.: FABMAS: an agent-based system for production control of semiconductor manufacturing processes. In: Mařík, V., McFarlane, D.C., Valckenaers, P. (eds.) HoloMAS 2003. LNCS (LNAI), vol. 2744, pp. 258–267. Springer, Heidelberg (2003)
16. Mönch, L., Stehli, M., Zimmermann, J., Habenicht, I.: The FABMAS multi-agent-system prototype for production control of waferfabs: design, implementation, and performance assessment. Prod. Plan. Control **17**(7), 701–716 (2006)
17. Mönch, L., Stehli, M.: ManufAg: a multi-agent-system framework for production control of complex manufacturing systems. Inf. Syst. e-Bus. Manag. **4**, 159–185 (2006)
18. Mönch, L., Fowler, J.W., Mason, S.J.: Production Planning and Control for Semiconductor Wafer Fabrication Facilities: Modeling, Analysis, and Systems. Springer, New York (2013)
19. Orcun, S., Asmundsson, J.M., Uzsoy, R., Clement, J.P., Pekny, J.F., Rardin, R.L.: Supply Chain Optimization and Protocol Environment (SCOPE) for rapid prototyping of supply chains. Prod. Plan. Control **18**(5), 388–406 (2007)
20. Orcun, S., Uzsoy, R.: The effect of production planning on the dynamic behavior of a simple supply chain: an experimental study. In: Kempf, K.G., Keskinocak, P., Uzsoy, R. (eds.) Planning Production and Inventories in the Extended Enterprise: A State of the Art Handbook, Chap. 3, vol. 2, pp. 43–80. Springer, New York (2011)
21. Ovacik, I.M.: Advanced planning and scheduling systems: the quest to leverage ERP for better planning. In: Kempf, K., Keskinocak, P., Uzsoy, R. (eds.) Planning Production and Inventories in the Extended Enterprise: A State of the Art Handbook, Chap. 3, vol. 1, pp. 33–43. Springer, New York (2011)
22. Pal, K., Karakostas, B.: A multi agent-based service framework for supply chain management. Procedia Comput. Sci. **32**, 53–60 (2014)
23. Roundy, R.: Report on practices related to demand forecasting for semiconductor products. Technical report. School of Operations Research and Industrial Engineering Cornell University (2001)
24. Schneeweiss, C.: Distributed Decision Making, 2nd edn. Springer, Berlin (2003)
25. Scholten, B.: The Road to Integration: A Guide to Applying the ISA-95 Standard in Manufacturing. ISA, Research Triangle Park, Durham (2007)

26. Van Belle, J., Philips, J., Ali, O., Germain, B.S., Van Brussel, H., Valckenaers, P.: A service-oriented approach for holonic manufacturing control and beyond. In: Borangiu, T., Thomas, A., Trentesaux, D. (eds.) Service Orientation in Holonic and Multi-Agent Manufacturing Control. SCI, vol. 402, pp. 1–20. Springer, Heidelberg (2012)
27. Van Brussel, H., Wyns, J., Valckenaers, P., Bongaerts, L., Peeters, P.: Reference architecture for holonic manufacturing systems: PROSA. Comput. Ind. **37**(3), 225–276 (1998)
28. Wobcke, W., Desai, N., Dignum, F., Ghose, A., Padmanabhuni, S., Srivastava, B.: What can agent-based computing offer service-oriented architectures, and vice versa? In: Desai, N., Liu, A., Winikoff, M. (eds.) PRIMA 2010. LNCS, vol. 7057, pp. 1–10. Springer, Heidelberg (2012)
29. Yoon, H.J., Shen, W.: A multiagent-based decision-making system for semiconductor wafer fabrication with hard temporal constraints. IEEE Trans. Semicond. Manufact. **21**(1), 83–92 (2008)