Chapter 8
State of the Practice and Future Needs for Production Planning and Control Systems

In this chapter, we describe information systems for production planning and control of wafer fabs. We start by discussing the current state-of-the-practice systems. Then we derive requirements for advanced production planning and control systems based on the results of the previous chapters. Next, we describe MES core functionality related to production control. Because the scheduling and dispatching functionality of many MESs is not adequate, we continue by describing specific dispatching systems. We also provide basic principles of a coupling architecture that allows for functionality extensions of MES in a plug-and-play manner. Moreover, we provide details of an agent-based architecture for modern production control systems. Software agents turn out to be an appropriate concept to implement enterprise-wide distributed production planning and control systems. We briefly discuss MCS functionality. Because of the increasing importance of production planning functionality, we finally describe requirements for ERP systems and APSs. These systems are usually provided by commercial software packages. Therefore, we also describe how these systems can interact with internally developed software components.

8.1 Motivation and State of the Art

In the previous chapters, different production planning and control methodologies were introduced. These algorithms have to be embedded into appropriate information systems because human decision makers are often not able to solve the corresponding decision problems since most of the algorithms proposed in the previous chapters require intensive computational effort and power.

Therefore, in this chapter, we mainly discuss the automated parts of the enterprise-wide information system, namely different application systems for production planning and control in wafer fabs or more generally in supply networks for semiconductor manufacturing. Enterprise-wide information systems

L. Mönch et al., *Production Planning and Control for Semiconductor Wafer Fabrication Facilities*, Operations Research/Computer Science Interfaces Series 52, DOI 10.1007/978-1-4614-4472-5_8,

for manufacturing contain so-called front-end and back-end application systems. Front-end systems are business-related systems, whereas back-end application systems are formed by shop-floor systems (see Qiu and Zhou [255]). Typical front-end application systems are used for sales management, customer services, and planning, while back-end application systems are used for supportive logistics, machine controls, and MES. It is essential for semiconductor manufacturers to invest in modern APSs and advanced MESs [158, 255] to be competitive. Semiconductor manufacturers can increase their revenues by providing services and sales through efficient channels supported by front-end application systems. At the same time, the manufacturing-related costs can be reduced by efficient back-end application systems.

We start by reconsidering the PPC hierarchy shown in Fig. 2.5 in Chap. 2. In Table 8.1, we assign corresponding front-end and back-end application systems to the different levels of this hierarchy.

Table 8.1: Assignment of application systems to the levels of the PPC hierarchy

Level	Application system
Planning	APS ERP Custom planning system
Order release	ERP/APS/MES
Scheduling	MES Custom scheduling system
Dispatching	MES MCS Custom dispatching system

Note that the individual software systems for production planning and control are typically extensions of commercial software packages that offer a more specialized functionality. In many companies, the packaged software systems provide the necessary data for the tailored software systems.

We start by developing a high-level picture of the architecture of an enterprise-wide information system and will refine the overall picture in subsequent sections. We show the overall architecture for an enterprise-wide information system in a wafer fab in Fig. 8.1.

Application systems of the PS other than the ERP system are often components of APSs. The MES communicates with the immediate control systems for the machines and the AMHS via middleware. Sometimes the notion of an enterprise service bus is used for this kind of middleware. The middleware is responsible for transaction processing and event handling. It also provides recovery functionality. Note that we avoid discussing manufacturing automation details in this monograph. We refer to Lee [158] for background material on this topic. Because the data flow between the planning systems

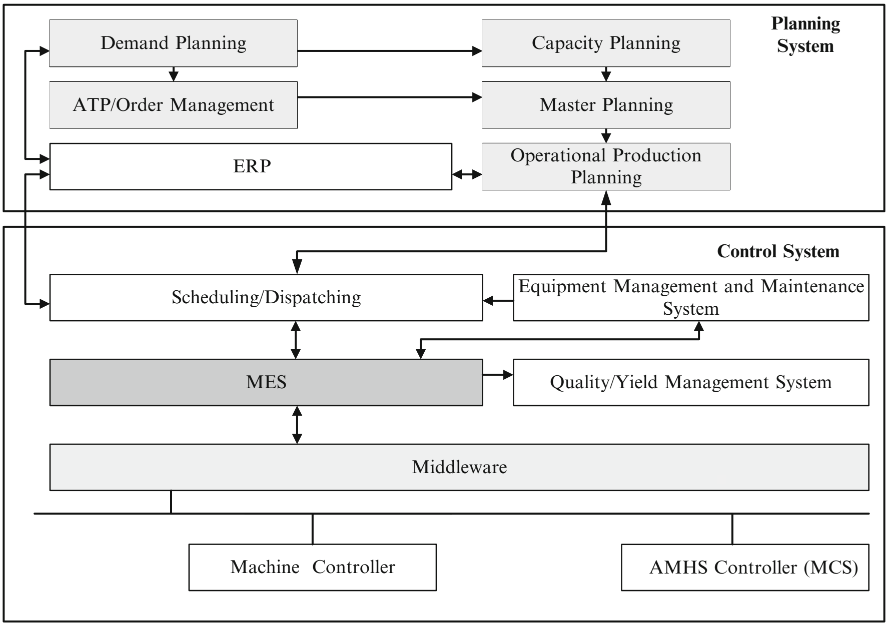

Figure 8.1: Enterprise-wide information system architecture

and the control systems is not extensive, proprietary application programming interfaces or web services are typically used to allow a message transfer between the planning systems and the MES.

When new business requirements appear, the supporting business processes have to change accordingly. Changing the business processes leads to new requirements for the enterprise-wide information system. In the next section, we will discuss some requirements for information systems that allow for these frequent changes.

8.2 Requirements of Production Planning and Control Systems

We derive several requirements for the different components of an enterprise-wide information system in wafer fabs or more generally in semiconductor supply chains. We identify the following requirements according to Qiu and Zhou [255]:

- The application systems should be able to deal with the data volume and the number of transactions that are typical for semiconductor manufacturing. This requirement is important because of the amount of data collected from the machine controllers and AMHS controllers.

- There are requirements related to the responsiveness of the application systems. A close to real-time response is critical for the MES, because the MES has to communicate with the machine controllers and AMHS controllers.
- The application systems should offer standardized connectivity with other application systems inside and outside the wafer fab. A plug-and-play connectivity that is suitable for different computing environments is highly desirable.
- Scalability and reconfigurability are key features of the application systems that are necessary to support business dynamics and growth.
- There is a trend towards multi-product facilities. The application systems have to be able to deal with many products. Furthermore, in some cases a single job is produced in several wafer fabs, i.e., borderless wafer fab scenarios (cf. Qiu and Zhou [255] and Sect. 2.2.2). The different planning systems and the MES have to support these requirements.
- The MES should allow an even tighter integration with automation solutions on the shop floor (see Lee [158]). Furthermore, it is anticipated that the advanced process control (APC) and AMHS operations will in the future be more tightly integrated with scheduling functionality. It is also expected that the importance of scheduling systems will be increased with a higher level of automation in future-generation wafer fabs. The MES architecture has to take these requirements into account.
- Appropriate production planning and control functionality should be offered by the application systems with respect to solution quality and the time needed to compute the solutions. It should be possible to modify and change the algorithms used quite easily to quickly react to changing business needs.
- The application systems should have an open and modular system architecture that is web-based to support the required interoperability for heterogeneous application systems. Furthermore, the target architecture should ensure a certain degree of adaptability for more advanced new technologies. We see also a trend towards collaborative manufacturing [47, 255]. A future enterprise-wide architecture has to take this trend into account.

It seems that service-oriented architectures (SOA) support many of the desired requirements. SOA proposes to define the required business functionality by a set of composable services. When needed because of changes in the business processes, new services can be obtained by service composition to support these business processes. However, it is difficult to assess the benefit of this type of architecture in semiconductor manufacturing.

It seems that SOA principles are used so far only on a preliminary basis in semiconductor manufacturing. An application on the factory automation level is described by Qiu [253]. A service-based application system for virtual manufacturing systems that involves several vendors in the semiconductor industry is discussed by Cherbakov et al. [47].

An interesting discussion of future architectures for enterprise-wide information systems in the pre-SOA era can be also found in Qiu [252]. In this context, E-manufacturing is defined as advanced manufacturing that takes advantage of the Internet and advanced information technologies to integrate the different manufacturing-related applications within a semiconductor supply network.

8.3 Production Control Systems

In this section, we start by discussing MES functionality. Then, we describe dispatching systems and explain their interaction with the MES. We present design and implementation details of a hierarchically organized multi-agent-system for production scheduling. Finally, we briefly discuss MCS functionality.

8.3.1 MES Core Functionality

An MES is an application system that consists of a set of integrated hardware and software components that are used to manage the production from job release until job completion [178, 182]. An MES is between the production PS and the execution of the production; it bridges the BS with the PS. The embedding of an MES into the overall system landscape of a wafer fab according to Chung and Jeng [50] is shown in Fig. 8.2.

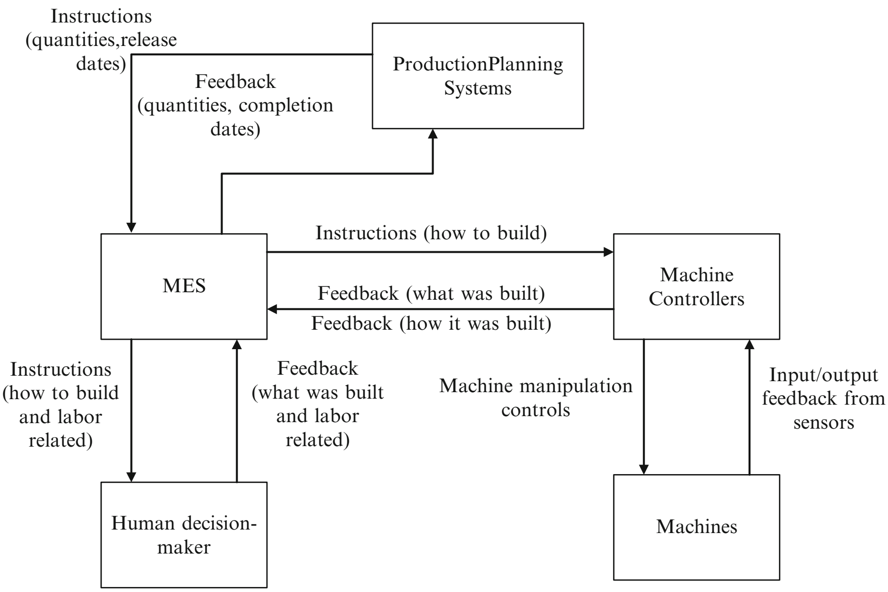

Figure 8.2: Interaction of an MES with other application systems

The main functionality of an MES in a wafer fab includes the following

- Equipment definition
- Product process definition
- Resource status tracking
- WIP status tracking
- Information management, i.e., data collection and acquisition
- Dispatching and scheduling
- Performance analysis

The equipment definition is used to define the processing capabilities of each machine. Usually, the machines are organized in types that are based on the associated processing capabilities.

The product process definition refers to the definition of the process flows. Each product type can potentially be manufactured using different versions. Each process version consists of several subroutes. Each subroute contains several consecutive process steps. A single process step refers to a recipe that is executed on the machine that is associated with the recipe. Each process step can have several versions. They depend on the machine and parameter calibrations. Current and planned production activities are used to define product processes.

Resource status tracking determines what task each single machine is currently performing. Furthermore, the current status of secondary resources is also tracked. Note that the status of the different resources involved in the AMHS is usually tracked in the MCS and not in the MES. The WIP status tracking refers to the situation where the progress of jobs is monitored. The aim is to create a full history for each working object of the BP. The information management functionality deals with monitoring and gathering of data about the different objects within the BS and the BP. Information management is an important prerequisite for implementing the resource and WIP tracking capabilities.

Dispatching and scheduling are among the key functionalities offered by an MES. They are based on the data that is determined by the MES. Some packaged MES products for wafer fabs offer dispatching functionality, but often additional off-the-shelf or homegrown solutions are used (cf. Pfund et al. [234] for the results of a corresponding survey). Therefore, we discuss dispatching and scheduling systems in more detail in subsequent sections.

The performance analysis component of an MES is responsible for comparing performance measure values with the corresponding objectives that are set by the management of the wafer fab or by customers. Graphical and numerical reports are also provided by this component.

Note that quality management and maintenance management are also key capabilities of MES [178, 182]. However, due to their high importance in semiconductor manufacturing, the corresponding functionality is often offered by separate application systems (cf. also Fig. 8.1). Therefore, we will discuss some of these systems briefly in subsequent sections.

Important packaged MES products in semiconductor manufacturing are WorkStream, WorkStream DFS, i.e., FAB300, from Consilium; PROMIS from Brooks Automation, Inc.; and SiView from IBM (see Qiu and Zhou [255]). The development of a homegrown MES is generally supported by the SEMATECH CIM-Framework [279]. However, it seems that packaged MES solutions have become the dominant implementations in wafer fabs (see Pfund et al. [234]).

8.3.2 Dispatching Systems

Dispatching systems are in place in most wafer fabs [234]. In an ideal scenario, a proposed dispatching method's superiority is established through experimental testing using actual semiconductor manufacturing data. In this scenario, the actual BS and BP data obtained in several relational databases including the MES database are extracted from the MES for use in developing and testing dispatching approaches. Therefore, a central repository is used to store this data. The application layer of the dispatching system interacts with the repository and also directly with the MES. This layer contains tools to construct blended, multilevel, conditioning, and truncated dispatching rules (cf. Sect. 4.1) and can be considered as a rule-based system (cf. Sect. 4.7.1). A simulation model of the current BS and BP is built using the data in the repository, and simulation analysis is used to assess the performance of the proposed dispatching rules that are developed using the functionality of the application layer. The dispatching system is completed by a graphical user interface (GUI) and a reporting interface.

Some manufacturers use dispatching systems that communicate directly with their MES in near real time. The dispatching list shown in the GUI of the dispatching system is implemented using functions of the MES.

Applied Materials' real time dispatcher (RTD) product [10] and the FabTime product [77] are commonly used [9, 238]. Once the efficacy of the dispatching approach is confirmed, the final step in this process is the deployment of the dispatching approach in an actual wafer fab.

RTD is a real-time, high-performance dispatching solution that helps manufacturers develop customized rules and improve dispatching analyses and decisions. As a member of Applied Materials' productivity family, RTD directs pre-staging, releases jobs, and adjusts production equipment loads through "what next, where next, and when next" rules. The goal of these rules is to improve the utilization of wafer fab equipment, carriers, and wafer fab personnel. RTD allows planners to define job selection logic in various rules and make production dispatching decisions in real time. This is facilitated by collecting and storing data from multiple sources in a high-speed, temporal repository. In this way, RTD is able to rapidly access both current shop-floor status and historical information over a desired definable time period [10].

FabTime is a web-based digital dashboard system that was created for the original purpose of helping wafer fabs to measure and improve their CT performance. In real time, FabTime provides a comprehensive view of everything a wafer fab manager would need for proactive CT management. The system includes over 120 standard wafer fab management charts focused on metrics such as WIP, moves, job age, WIP turns, per-visit CT, factory CT, machine states, machine overall equipment effectiveness (OEE), scrap, and holds. The FabTime system works by taking a continuous feed of operational transactions from the wafer fab's MES and processing these transactions in real time for storage in a database. Both chart and tabular output can be viewed using a standard web browser across a corporate intranet. In addition to providing real-time alerting so that users can specify conditions under which to be notified (such as a machine down event or the release of a new product), FabTime also includes optional dispatching and static capacity planning modules. As the FabTime system maintains real-time knowledge of current wafer fab status through its database information (which is constantly updated by the MES), FabTime's dispatching module [77] can effectively promote in-fab deployment of the system's recommended job dispatch list for any machine in the wafer fab.

The typical architecture of a dispatching system is shown in Fig. 8.3. It is similar to those described in Sect. 3.3.2 because the simulation model represents the BS and the BP in both situations.

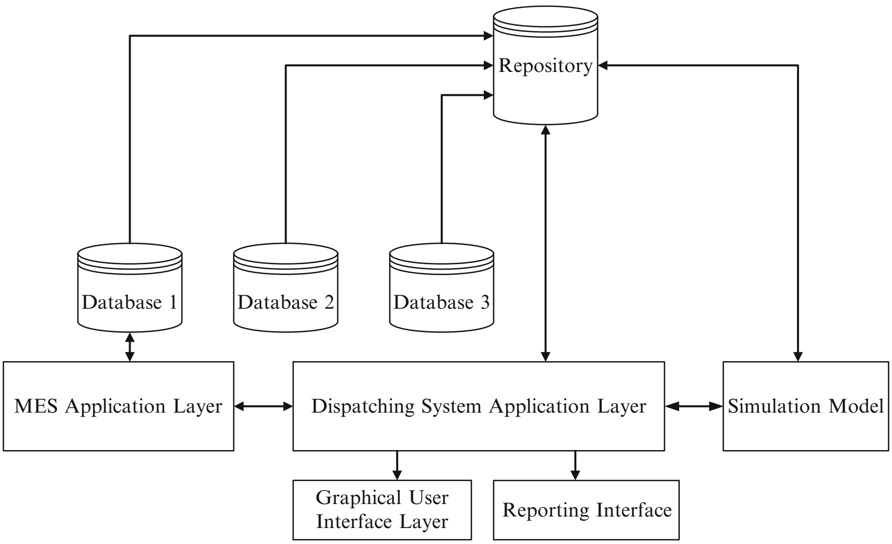

Figure 8.3: Architecture of dispatching systems

8.3.3 Scheduling Systems

Similar to dispatching systems, scheduling systems often are not part of the MES because it is hard to adapt generic scheduling functionality to the situation in a specific wafer fab. Currently, wafer fab-wide scheduling systems are generally not in use (see Mönch et al. [207]). However, scheduling systems for work centers or even work areas are in use in many wafer fabs. Some example systems are described in [28, 147, 329, 330].

We continue by describing the architecture of scheduling systems. A scheduling system typically consists of the following components (see Framinan and Ruiz [90]):

- User interface component
- Schedule generator component
- Business logic component
- Database management component

The main functionality of the different components can be characterized according to Framinan and Ruiz [90] as follows. The user interface component offers the required interfaces between the user of the scheduling system and the system itself. The component allows for the representation of scheduling output. Finally, the user can initiate the scheduling process, including choosing appropriate parameter settings, using the user interface.

The schedule generator component contains all the functionality that is required to determine schedules for the users. It consists of an algorithm library that contains scheduling and rescheduling algorithms. An algorithm generator is responsible for adding new algorithms or obtaining new ones by combining existing ones from the algorithm library. The input data is transformed by a preprocessor into a format that is appropriate for the scheduling algorithm. Finally, a schedule is calculated.

The business logic component is between the user interface component, the schedule generator component, and the database management component. It ensures the required abstraction when data is accessed from the database by the user interface or by the schedule generator. The business logic component is a transformation component. Finally, the database management component is responsible for storing all the data that is obtained from the MES and from other databases similar to the repository in the case of dispatching systems. It provides interfaces to the application systems that contain the data that are necessary for scheduling.

In Sect. 8.3.4, we present some details of an agent-based scheduling system prototype that can be used in wafer fabs. We show how the NDSBH and the IDSBH algorithms from Sect. 5.4.6 can be implemented in a distributed manner.

8.3.4 FABMAS: An Agent-Based Scheduling System

Software agents allow for the implementation of distributed planning and control algorithms. The agents are able to act autonomously; on the other hand, their communication abilities ensure a cooperative behavior and the fulfillment of global system goals. Furthermore, agent-based systems facilitate maintenance and further development tasks of the software (see Weiss [319] and Wooldridge [324]).

We start by the agentification of the scheduling problem for a single wafer fab. Many approaches address the problem of identifying proper agents for a given application domain. We refer, for example, to the Gaia approach described by Zambonelli et al. [331]. The Gaia approach is a generic approach that assigns a set of roles to a given domain. We define a role as a class that determines the normative behavior repertoire of an agent (see Odell et al. [217]). Interactions are identified that take place between the different roles. However, as pointed out by Bussmann [38] and by Bussmann et al. [39, 40], it is necessary to analyze and understand the decisions in the course of the production control process.

We combine the approach of Bussmann [38] with the PROSA reference architecture for holonic manufacturing systems (cf. Van Brussel et al. [310]). PROSA is an abbreviation for *P*roduct, *R*esource, *O*rder, and *S*taff *A*rchitecture. The reference architecture suggests building agent-based production control systems by using these agent (holon) categories. Furthermore, PROSA provides a high-level description of the interaction of instances of these agent categories and a set of examples for using the architecture. A holonic manufacturing system is a system of holons that are able to cooperate to achieve a common objective (see McFarlane and Bussmann [179]). An autonomous and cooperative building block of a manufacturing system for transforming, storing, transporting, and validating information and physical objects is called a holon. Holons can be part of other holons, i.e., there is a recursive structure. Note that for our purposes, the difference between holons and agents is not important (cf. McFarlane and Bussmann [179] for a discussion of related issues). We consider three steps:

1. Analyze the decision-making process.
2. Identify the necessary agents.
3. Choose appropriate interaction protocols.

Based on the proposed hierarchical approach described in Sect. 5.4.6, we distinguish three types of decisions. These decision types are presented in Table 8.2.

The decision-making on the top and middle layer will be performed in a rolling horizon or event-driven manner. The ICA scheme (cf. Sect. 5.4.6) is used on the top layer, while NDSBH or IDSBH (cf. Sect. 5.4.6) makes decisions on the middle layer. The decisions of the base layer are made dependent on the situation of the machine group. If the schedule is infeasible, then

Table 8.2: Decision types in FABMAS

Layer of the hierarchy	Decision	Decision space
Top layer	Start and end dates for each macro operation of a job	Time slots for the start and end dates
Middle layer	Assignment and sequencing decisions for operations of a job	A specific machine among parallel machines, a concrete time slot on that particular machine
Base layer	Whether to follow the schedule or not, assignment and sequencing decisions for operations in the latter case	A specific machine among parallel machines, a concrete time slot on that particular machine

decision-making entities of the jobs make dispatching decisions together with the decision-making entities of the machine groups. A contract net-type allocation algorithm (cf. Weiss [319] for contract nets) is used.

Starting with PROSA, we distinguish between decision-making agents and staff agents. Decision-making agents solve decision problems while the staff agents try to support them in the course of the decision-making process. In PROSA, we find order, product, and resource agents as abstract classes. We identify eight decision-making agent types, i.e., roles, in our application scenario.

1. Each job agent represents a single job.
2. Batch agents are used to control a certain batch, i.e., a collection of jobs that are intended to be processed at the same time on the same machine.
3. A PM agent represents a preventive maintenance order.
4. Work center agents represent machine groups on the shop floor.
5. We aggregate several work center agents into one work area agent.
6. The fab agent consists of all work area agents.
7. Tool agents are used for the representation of auxiliary resources.
8. We consider technology agents that encapsulate the product knowledge, i.e., the routes, according to the product holons of PROSA.

We identify two additional staff agents that encapsulate the scheduling and monitoring functionality for the hierarchical production control scheme.

We summarize the basic functionality of the members of the decision-making and staff agency in Table 8.3. The different agent roles are shown in Fig. 8.4. Each role is described by a set of possible behaviors. A single behavior is given by a set of states and by transition paths from one state to another state. PROSA describes the basic interaction between product, resource, and order agent roles (see Van Brussel et al. [310]). For the purpose of FABMAS, a modeling of the interactions between decision-making and

Table 8.3: Functionality of decision-making and staff agents

Member of the agency	Task description
Fab agent	– Coordinating the work of the fab scheduling agent, the monitoring agent, and the work area agents
Fab scheduling agent	– Preparing to run ICA – Running ICA and providing scheduling information
Work area agent	– Coordinating the work of the corresponding work area scheduling and monitoring agent – Decision-making for choosing the proper machine criticality measure for NDSBH or IDSBH – Providing information services
Work area scheduling agent	– Preparing to run NDSBH or IDSBH – Running the heuristic – Providing scheduling information
Work center agent	– Implementing the work area schedules – Mediating in the case of a contract net-type allocation algorithm
Tool agent	– Implementing the work area schedules
Job agent	– Coordinating the processing of the job that is represented by the job agent – Negotiating with work center agents
Batch agent	– Coordinating the processing of the batch that is represented by the batch agent – Negotiation with work center agents
PM agent	– Coordinating the preventive maintenance step that is associated with the agent
Technology agent	– Providing routing information to job agents, work center agents, and various staff agents – Dynamically changing the assignment of work centers to process steps, e.g., caused by machine breakdowns

staff agents is more important. We identify four different behaviors for staff agents:

- PrepareSolution behavior
- ParameterizeAlgorithm behavior
- SolveOrInterrupt behavior
- CommunicateSolution behavior

A brief description of the different behaviors of staff agents is given in Table 8.4.

For decision-making agents, we identify the behaviors:

- PrepareDecisionMaking
- MakeDecision
- InformDecisionMakingAgents
- StartStaffAgent
- RequestDecisionMakingAgentResults

The different behaviors of decision-making agents are briefly described in Table 8.5.

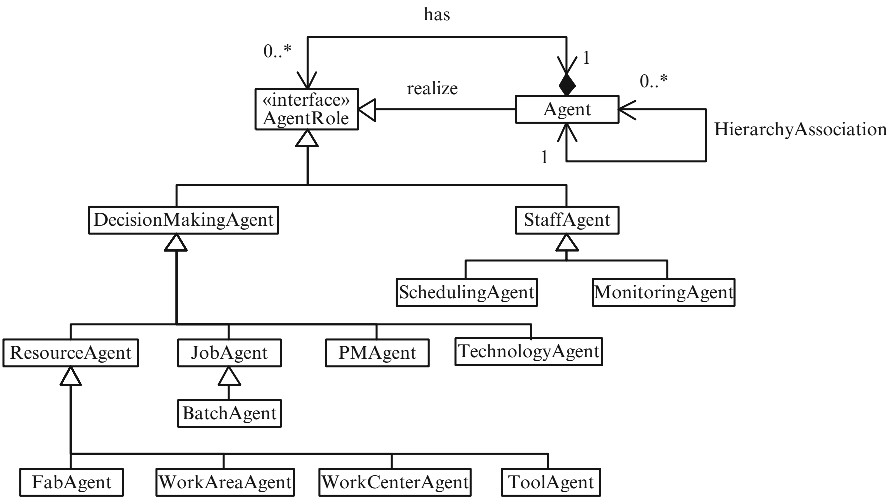

Figure 8.4: Agent roles in FABMAS

Table 8.4: Behaviors of staff agents

Behavior	Description
PrepareSolution	This behavior models the data collection and parameter determination phase of the problem solution process
ParameterizeAlgorithm	Parameters are assigned to the solution algorithm
SolveOrInterrupt	Feasible solutions are determined for the problem under consideration
CommunicateSolution	The behavior informs the decision-making agent associated with the staff agent

Next, we briefly discuss the implementation of the FABMAS prototype [204]. The use of an agent toolkit or framework was avoided because none of the existing tools allows for the implementation of hierarchies and the usage of discrete-event simulation for performance assessment. Furthermore, problems with respect to computational performance were expected because the majority of toolkits are Java-based. The prototype is implemented using the Microsoft .NET framework for inter computer communication. Within .NET, we implement the prototype in the C# programming language, while some parts of source code are developed in the C++ programming language. According to the foundation for intelligent physical agents (FIPA) [79] Abstract Architecture for agent systems, we develop an agent runtime environment. The runtime environment consists of an agent directory service, an agent management system, an agent container, and an agent communicator. These parts of the system provide services that are used by the agents to get information about other agents and to communicate and interact with them.

Table 8.5: Behaviors of decision-making agents

Behavior	Description
PrepareDecisionMaking	The agent is prepared for a decision-making process, for example, by data collection
MakeDecision	A decision is made by the agent
InformDecisionMakingAgents	Another decision-making agent is informed of a decision of the agent
StartStaffAgent	A service of a staff agent is requested by the decision-making agent
RequestDecisionMakingAgentResults	A certain result obtained by another decision-making agent is requested

For communication purposes, the agent communicator encapsulates communication capabilities. Each communication act between agents is handled by the agent communicator. The agent directory service is the location where agents register their specification as a service entry. Agents are able to ask the local directory service to find information about other agents they want to interact with. If the information is not available, the directory service tries to find the information by contacting other directory services within the whole multi-agent-system. Hence, it is not necessary to establish a global directory service as a centralized information point in a distributed system. Each agent runtime environment requires an agent management system that administers the life cycle of each agent. As a result, the management system is responsible for creating the agents, provides potential mobility services, and removes an agent if it is no longer needed. The last component of the agent runtime environment is the agent container as a collection of all active agents within the environment.

Next, we discuss the communication in FABMAS. Various opportunities for implementing communication abilities are given by choosing the .NET Platform for the development of the agent system. An agent communicator is part of every runtime environment and provides two capabilities for communication. The multicast communication is used for announcement of new active runtimes and by the agent directory service to keep their agent list up-to-date. The direct communication is used for communication purposes between single agents. Both types of communication are implemented by using the .NET Remoting framework for distributed computing.

The agent communicator hides all the communication capabilities from the agents and can be used as an interface. An agent that is intended to send a message to another agent transfers the message to the agent communicator. The communicator determines the location of the agent, and, if the other agent is on a remote host, .NET Remoting is used for sending the message. If the receiver agent is in the same runtime, the message is directly put into the mailbox of the receiver agent.

The agent communicator is implemented as client and server, simultaneously. The usage of threads makes the agent communicator concurrently executable. According to the FIPA proposal for an agent communication language, we use the content format described by Mönch and Stehli [195]. The format is basically given by a context-free grammar, and it relies heavily on the ontology described by Mönch and Stehli [196].

We continue by discussing the representation of hierarchies within FABMAS. The hierarchy according to the suggested hierarchical approach is modeled by using an agent identifier. The identifier is a pointer to agents. An agent identifier encapsulates the agent name, the address where the agent is located, and the services provided by the agent. Each agent on a higher level stores the agent identifier of its child agents on the next hierarchy layer. The fab agent contains all agent identifiers of the work area agents. Each work area agent knows the identifiers of the agents associated with work centers that are part of the work area. On the other hand, each work center agent knows its work area agent. Thus, a structure exists that allows for communication and cooperation among the decision-making entities at the same hierarchy layer and between adjacent layers.

We use an extension of the architecture described in Sect. 3.3.2 to carry out the performance assessment of FABMAS. The center point of this architecture is a blackboard-type data layer that contains all the information to execute the ICA heuristic (cf. Sect. 5.4.6), construct the disjunctive graphs for the NDSBH and IDSBH schemes (cf. Sect. 5.4.6), and make the scheduling decisions. The data layer is between a simulation model that emulates the manufacturing process of interest and the FABMAS system. The objects of the data layer are updated in an event-driven manner by appropriate simulation events. Calculated schedules are submitted to the simulation engine AutoSched AP in order to use the information of the schedules in a dispatching-based manner. The architecture allows for rolling horizon-type scheduling as well as for event-driven rescheduling activities. We implement an interfacing component in order to connect FABMAS with the blackboard. The .NET system supports the encapsulation process and also registers the component for the Windows Registry. The .NET framework creates a component object model (COM) callable wrapper during runtime. This wrapper offers user-specific interfaces to the component and also the typical interfaces of COM. The blackboard can call methods of the interface and transfer data via parameters to the FABMAS multi-agent-system. This data is forwarded by the .NET Remoting component to the agent runtime environment. On the other hand, data is transported back from the multi-agent-system to the blackboard via method calls using reference parameters. The communication between the described components is shown in Fig. 8.5.

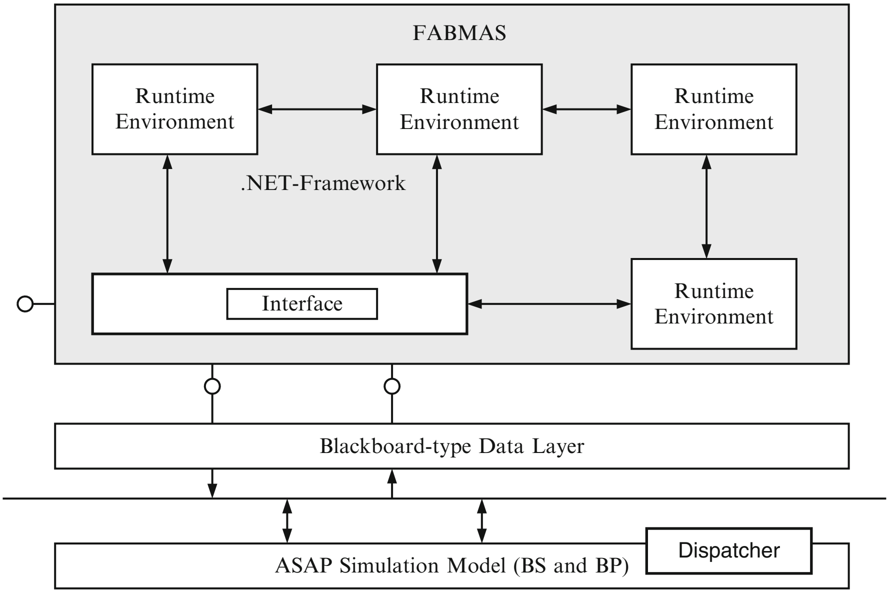

Figure 8.5: Architecture for performance assessment of the FABMAS system

We wish to point out that agent-based approaches have become popular in semiconductor manufacturing over the last decade. In addition to FABMAS, there is an agent-based scheduling system at AMD/GLOBALFOUNDRIES discussed by Pinedo [240]. In addition, Yoon and Shen [328] describe a multi-agent-system for wafer fabs that makes scheduling decisions based on a bidding scheme.

8.3.5 Additional Application Systems as Part of the Control System

Among the different controls, we discuss only the MCS because of its importance in modern wafer fabs. The MCS is responsible for initiating and coordinating movements of carriers for wafers and reticles (see Foster and Pillai [82]). Furthermore, it coordinates interbay and intrabay activities between stockers and machine load ports. Intrabay and intrabay AMHS components are monitored by the MCS. Carrier locations within the AMHS are stored in the MCS.

The MCS interfaces with the MES and also with different components of the AMHS. It receives job destination information and also dispatching- and scheduling-related information from the MES. The MES is updated by the MCS when jobs are moved within the wafer fab. The MCS receives the identification numbers of jobs and carriers when they enter the AMHS. The movement of jobs is instructed and coordinated by the MCS. Finally, the MCS collects information with respect to job movements, job locations, and AMHS reliability.

Equipment engineering systems (EES) are used as tools for vendors to monitor process control of machines and tune process parameters remotely (see Lee [158]). It is a physical implementation of equipment engineering capabilities (EEC). The main goal of an EES is to increase the OEE. An EES contains fault detection and classification (FDC) and predictive maintenance. Sometimes, the functionality of a quality/yield management system is also part of an EES. FDC aims to detect anomalies in process control that can cause large quality problems, classify the problems, and report them. Statistical methods and methods from data mining and machine learning are applied to solve this task. Predictive maintenance starts by detecting a failure symptom. Then, the residual life of the equipment is predicted before the failure occurs. A failover process is finally initiated. An equipment management and maintenance system (EMMS) is a software tool that monitors and tracks equipment states and preventive maintenance schedules in real time. It is similar to an EES.

Quality requirements have become stricter over the years in semiconductor manufacturing (see Lee [158]). Therefore, separate quality/yield management systems are installed in most wafer fabs, and they are not generally part of the MES. APC functionality is an integrated part of a quality/yield management system. APC includes run-to-run control, FDC, statistical process control (SPC), and virtual metrology (VM). Run-to-run control is responsible for adapting process control parameters (see Moyne et al. [211]). Therefore, measurements of process sensors are used that are taken on a wafer-to-wafer, job-to-job, or batch-to-batch basis to adjust recipes. Typically, statistical methods and methods from machine learning are used. FDC is typically part of the EES. SPC is used to monitor the physical measurements of the wafers. This measurement is carried out using metrology equipment. VM is based on the insight that metrology is time- and cost-consuming. Therefore, the number of wafers that can be measured is limited. VM uses mathematical modeling to estimate the values of metrology measures on the wafers depending on FDC indicators without physical metrology operations.

While we have described only production control-related application systems so far, in the following section, we discuss in more detail planning-related application systems. Such systems provide instructions that are important for production control-related application systems.

8.4 Production Planning Systems

In this section, we describe the main functionality of ERP systems and APSs. We then briefly discuss the interaction with other systems.

8.4.1 ERP and APS Core Functionality

We start by describing the core functionality of an ERP system. ERP systems are typically transaction-oriented software packages. They offer functionality related to finance, human resources, manufacturing and logistics, and finally sales and distribution (see Hopp and Spearman [119]). ERP systems are operational application systems. They store important BS- and BP-related data, called master data. This includes product data, bills of materials, routes, and resource- and job-related data. ERP systems are often organized in different software modules. Each of these modules supports one of the main ERP functionalities. ERP systems follow an integrated approach that is characterized by the following features [119]:

- Integrated functionality
- Integrated databases
- Consistent user interfaces
- Unified architecture and tools to maintain, improve, and extend the system
- Single vendor/contract and a unified product support

The manufacturing- and logistics-related module typically offers MRP and manufacturing resources planning (MRP II) functionality. However, it is well known that several assumptions of these approaches, like infinite capacity and fixed CT, might lead to fundamental problems and low performance of manufacturing systems.

Because of the simple or even missing bill of materials in many wafer fabs, the production planning functionality of ERP systems is used only to a small extent in wafer fabs. The order management functionality offered by ERP systems is often the most important functionality in this context. The second important functionality is demand planning, i.e., forecasting. However, on the entire enterprise level, ERP systems are often complemented by APSs.

An APS supports decisions within the supply chain management context on a long-term, mid-term, and finally short-term planning level. They can be considered as extensions of ERP systems because these systems are typically not able to solve the entire set of decision-making problems associated with a supply chain. An APS is an application system that supports production planning tasks using OR and AI methods taking the finite capacity of the BS into account. The most important features of an APS are according to Fleischmann et al. [81] as follows:

- Integrated planning along the entire supply chain
- Optimization-based approaches based on mathematical models and algorithms that either provide optimal or heuristic solutions

- Application of a hierarchical planning approach that decomposes the entire planning problem in a series of smaller, less complex subproblems that are assigned to the different layers of the hierarchy

Similar to ERP systems, APSs are usually offered as packaged software systems. They provide functionality related to the strategic design of the manufacturing network, demand planning, supply network planning, external procurement, production planning and scheduling, transportation planning and vehicle scheduling, order fulfillment, and available-to-promise (ATP). In contrast to ERP systems, APSs often use algorithms that are based on data in the memory of the computers. This approach, sometimes called LiveCache technology, allows for very fast accessing of the data, because accessing relational databases is avoided.

Next, we describe how the different subsystems of the overall PS in Fig. 8.1 are implemented in semiconductor manufacturing. We start with demand planning. Demand planning is related to the task of forecasting the future market demand for the semiconductor products of the companies. Demand planning is more important when the company follows a make-to-stock rather than a make-to-order strategy. Long-term demand forecasts are important for the design of the supply network and for capacity expansion decisions. Mid-term forecasts are essential for the coordination of procurement, manufacturing, and distribution. They form an important input for supply network planning, i.e., master planning. Finally, short-term demand forecasts are essential to ensure high service levels (see Günther [109]).

Based on a survey by Roundy [271], we conclude that the demand planning functionality of ERP systems and APSs is often used in semiconductor manufacturing. However, homegrown solutions are also widespread in this industry. A prototype that offers optimization-based short-term forecast functionality in semiconductor manufacturing is described by Mönch and Zimmermann [199]. It uses web services to interact with different application systems to gather the required data.

The main purpose of an order management component consists in matching customer orders against quantities available in stock or from already planned production orders. The investigation of whether a delivery can be performed or not is called ATP. Usually, one looks for available stock that can be promised for delivery. Many APSs are able to check the available capacity that can be used to place new orders. Furthermore, it is also checked whether or not the size of already planned orders can be increased. This feature is called capable-to-promise (CTP). An order management component is also responsible for order entry and customer service, reporting, order processing, and financial processing.

Capacity planning decisions are described in Sect. 7.3. It seems that often homegrown decision-support systems, including a commercial MIP solver, are used to make these kinds of decisions. Simple spreadsheet-based applications that use the included LP or MIP solvers are also popular. One example for the first class discussed in the literature is CAPS (see Bermon and Hood [24]) that was run at several IBM wafer fabs to support capacity planning decisions (cf. Sect. 7.3).

Master planning in semiconductor manufacturing is discussed in Sect. 7.2. Master planning algorithms are typically provided by APSs; however, because packaged APSs are often not able to deal appropriately with process restrictions of semiconductor manufacturing, homegrown solutions are also in use. In Kallrath and Maindl [135], it is indicated that heuristics from SAP APO are used for master planning in semiconductor manufacturing.

Operational planning is short-term planning that is discussed in Sect. 7.1. It is often supported by homegrown, spreadsheet-like solutions. In a certain sense, the ICA algorithm, presented in Sect. 5.4.6, is somewhere between operative planning and production control. Usually, this kind of approach is provided by an MES.

A discussion of some trends in planning systems for supply chains in the semiconductor manufacturing industry can be found in Banerjee [20]. Some empirical evidence of APS failures in semiconductor manufacturing can be found in Lin et al. [164]. It is shown that APSs in semiconductor manufacturing often do not perform better than humans that make planning decisions with computer support.

8.4.2 Interaction with Other Systems

We can see from Fig. 8.1 that there is a link between operational planning systems and dispatching and scheduling systems. Furthermore, there can be a link between the ERP system and the MES because the MES needs order information. In addition, when a job is completed, this information has to be sent to the ERP system.

Within the PS, there are several possibilities for the interactions of APS and ERP systems. In the simplest case, there is exactly one APS that is on top of the ERP system. But it is also possible that several APSs are in use together with one centralized ERP system. Furthermore, we might have one centralized APS and several ERP systems. Typically, global supply chains in the semiconductor industry contain multiple APSs and several ERP systems. Little is known about how SOA-type approaches will impact the architecture of the next-generation PS in semiconductor manufacturing.