Design and Implementation of a Service-based Scheduling Component for Complex Manufacturing Systems

Lars Mönch
Department of Mathematics and Computer Science, University of Hagen, Universitätsstraße 1, 58097, Hagen, Germany

Keywords: Service-oriented Computing, Scheduling, MES, Complex Manufacturing Systems.

Abstract: Scheduling is highly desirable in complex manufacturing systems. However, there is still a mismatch between academic scheduling research, the scheduling solutions offered by software vendors, and the requirements of real-world scheduling applications. In this paper, we describe the design and the development of a scheduling component prototype that is based on web services. It exploits the idea of a hierarchical decomposition of the overall scheduling problem allowing the integration of different problem-specific scheduling algorithms for sub-problems. We discuss how appropriate services can be identified and implemented and how the resulting scheduling component can be used to extend the functionality offered by manufacturing execution systems (MESs).

SCIENCE AND TECHNOLOGY PUBLICATIONS

1 INTRODUCTION

This research is motivated by scheduling problems that are found in complex manufacturing systems, as for example, semiconductor wafer fabrication facilities (wafer fabs). Complex manufacturing systems are characterized by a diverse product mix, many machines, a large number of jobs, sequence-dependent setup times, and batching. Here, batching means that several jobs can be processed at the same time on the same machine. Scheduling is challenging in such an environment. However, it is highly desirable because of the increasing automation pressure. In contrast to previous papers (cf. Mönch and Driessel, 2005), we are not interested in proposing a new scheduling technique. Instead of this, we deal with the question of how to design scheduling components from a functional and also from a software technical point of view. It turns out that the data available in Enterprise Resource Planning (ERP) systems and Advanced Planning Systems (APS) are not fine-grained enough to allow for making detailed scheduling decisions. Furthermore, their actuality with respect to time is not appropriate. MESs are a natural carrier of scheduling functionality (McClellan, 1997; Meyer *et al.*, 2009). However, the scheduling capabilities of packaged MESs are often not appropriate because they are too generic (cf. Pfund *et al.*, 2006 for the results of a survey of the acceptance of packaged scheduling solutions in the semiconductor manufacturing industry). In this paper, we research the problem of designing a scheduling component that can be used by an MES. In a certain sense, this paper extends previous work carried out for the ERP domain (cf. Mönch and Zimmermann, 2009). The design of the component is derived taking an appropriate hierarchical decomposition of the overall scheduling problem into account. After identifying appropriate services, we implement a prototype based on web services. Such questions are rarely discussed in the literature so far (cf. Framinan and Ruiz, 2010 for a recent survey of the architecture of scheduling systems).

The paper is organized as follows. In the next section, we describe the problem and discuss related literature. We present the hierarchical decomposition of the overall scheduling problem in Section 3. Furthermore, we describe how appropriate services can be indentified. The implementation of the prototype is described in Section 4. We discuss also some limitations of the proposed approach and future research needs.

2 PROBLEM DESCRIPTION

2.1 Problem

In current MESs for complex manufacturing systems,

Mönch L..
Design and Implementation of a Service-based Scheduling Component for Complex Manufacturing Systems.
DOI: 10.5220/0004002702840290
In *Proceedings of the 14th International Conference on Enterprise Information Systems* (ICEIS-2012), pages 284-290
ISBN: 978-989-8565-10-5
Copyright © 2012 SCITEPRESS (Science and Technology Publications, Lda.)

dispatching functionality often is offered instead of the more sophisticated scheduling functionality. Optimization kernels are typically based on genetic algorithms or on generic commercial constraint programming and mixed integer programming libraries (cf. Fordyce *et al.*, 2008). Scheduling systems are mainly developed only for parts of the manufacturing system, for example for the leading bottleneck machine group. There is only little interaction of software vendors and academic research (cf. Kellogg and Walczak, 2008). There are several reasons for this situation.

1. Scheduling of production jobs is often combined with transportation scheduling, process planning, staff scheduling, and finally advanced process control decision-making.
2. Global scheduling systems fail because humans on the shop floor are not involved in the resultant scheduling decisions. It seems that often the notion of reasonable automation is not taken into account.
3. Scheduling algorithms depend to a large extent on the objectives and constraints taken into account. That means that slight changes in the objectives and the constraints might lead to totally different algorithms. Dispatching rules strongly support this behavior by its inherent myopic view. Generic scheduling solutions have some limitations with respect to dealing with this situation. They are often not well accepted by people on the shop floor.
4. The data for scheduling decisions is located in different operative application systems. MES- and Material Control System (MCS)-related data are very important in this context.
5. Supplying appropriate data to the scheduling algorithms is important. However, scheduling algorithms that take many details into account require at the same time data that is fine-grained.

Analyzing these insights results in the conclusion that striving for a more detailed modeling is inapplicable because a more detailed consideration of constraints leads to sophisticated algorithms and also to a more difficult data supply. Therefore, it seems important to focus on the quintessence of scheduling, i.e., considering the finite capacity of the manufacturing system is more important. However, it is possible to deal with the finite capacity on a more aggregated level.

In this paper, we address the question of how a scheduling component has to be organized to take this vision into account. Therefore, the design of a service-based scheduling component is discussed that is based on an appropriate distributed hierarchical decomposition of the overall scheduling problem. The resultant design is validated by a prototypical implementation of such a component.

2.2 Related Work

A web service-based specification and implementation of ERP components is described, for example, by Brehm and Marx Gomez (2007) and Tarantilis *et al.* (2008). However, a direct application of these ideas to scheduling is not possible because of the different level of detail. A conceptual proposal for an MES based on web services that can be used in small- and medium-size enterprises in Mexico is discussed by Gaxiola *et al.* (2003). But again, no specific details of possible scheduling functionality are included in this paper. This is also true for the recent survey paper by Framinan and Ruiz (2010), where the usage of web services is only mentioned, but not further elaborated.

A service-oriented integration framework for complex manufacturing systems is presented by Qiu *et al.* (2007). A certain portion of a traditional MES, especially with respect to feedback from the shop floor, is implemented within the framework, but again, scheduling functionality is not covered.

There is some work done for the identification of services (cf. Winkler and Buhl, 2007 for the financial domain and Mönch and Zimmermann, 2009 for ERP-related services). However, to the best of our knowledge, there is no work available that addresses this question for scheduling services in complex manufacturing systems. A distributed scheduling system for complex job shops based on software agents is presented in Mönch *et al.* (2006). But in contrast to web services, software agents and multi-agent-systems are still not widely accepted in applications on the shop floor. In this paper, we will show that some of the scheduling functionality described by Mönch *et al.* (2006) can be provided using principles of service-oriented computing.

3 IDENTIFICATION OF APPROPRIATE SERVICES

3.1 Distributed Hierarchical Decomposition

As discussed in Subsection 2.1, we are interested in making manufacturing system-wide scheduling decisions without increasing the level of detail for modeling. This goal is mainly reached by an appropriate hierarchical approach.

We start by describing the assumed physical decomposition of the base system of the manufacturing system. The routing of the jobs, the dynamic entities in the system, takes place between different groups of parallel machines. Parallel machines offer the same functionality in a manufacturing system. A single group of parallel machines is called a work center. The work centers that are located in the same area of the manufacturing system are aggregated into work areas. On the highest level, we find the entire manufacturing system, i.e., the different work areas form the base system. In order to solve the overall scheduling problem, often hierarchical decomposition approaches have been applied.

In this paper, we consider a two-layer hierarchical approach. The resulting scheduling scenario is shown in Figure 1 as a Unified Modeling Language (UML) sequence diagram and will be explained in more detail below.

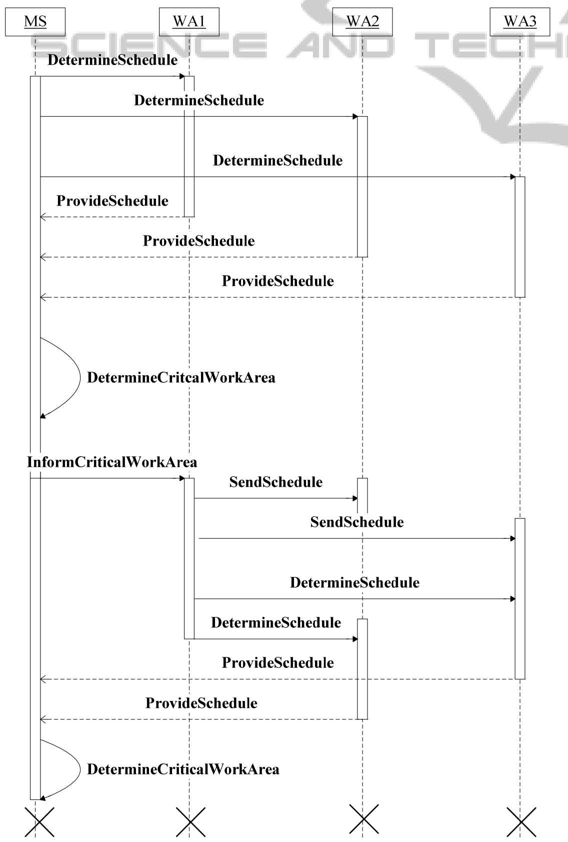

Figure 1: Sequence diagram for distributed scheduling.

In complex manufacturing systems, planned start dates and completion dates are determined with respect to a fixed work area (cf. Mönch and Driessel, 2005 and Mönch *et al.*, 2006) on the top-layer. Here, an aggregated model is used taking the capacity only on the work center level into account. Consecutive operations that are related to one work area are combined into macro operations. This results in aggregated routes that consist of these macro operations. The resulting start and completion dates can be used to set production goals for each decision-making entity, i.e., a scheduling unit, on a certain work area. This approach is called job planning to differentiate it from the more detailed scheduling. In Figure 1, we denote the decision-making entity of the entire manufacturing system by MS. The base-layer is formed by the different work area decision-making entities. We can see three work areas, denoted by WA, in Figure 1. This layer results in schedules for each single work area. Then, it determines a critical work area with respect to a criticality measure, for example, the tardiness of the jobs with respect to the work area where they are scheduled. Based on the schedule for the critical work area, updated ready dates and due dates are sent to the non-critical work areas and used to determine new schedules. This procedure is repeated for the next most critical work area in a recursive manner until the last work area is reached.

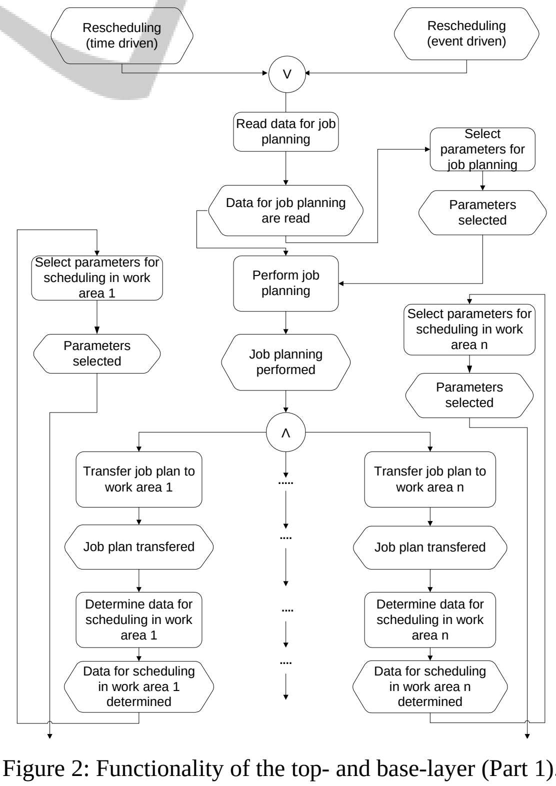

Figure 2: Functionality of the top- and base-layer (Part 1).

Figure 1 does not provide enough information from a process modeling point of view. Therefore, we provide event-driven process chains (EPC) for the researched scheduling scenario in Figure 2 and Figure 3.

These process models can be used as a starting point for identifying appropriate services, because service-oriented architectures (SOAs) usually distinguish between process, service, and technology models (cf. Siedersleben, 2007). Figure 2 shows the job planning step and major parts of the preparation phase for the scheduling activities on the base-layer. The iterative procedure that corresponds to the decisions on the base-layer is shown by continuing the EPC from Figure 2 in Figure 3. Only a single iteration is depicted in this figure for the sake of simplicity. Note that a simple process model is provided by the EPC in Figure 2 and Figure 3.

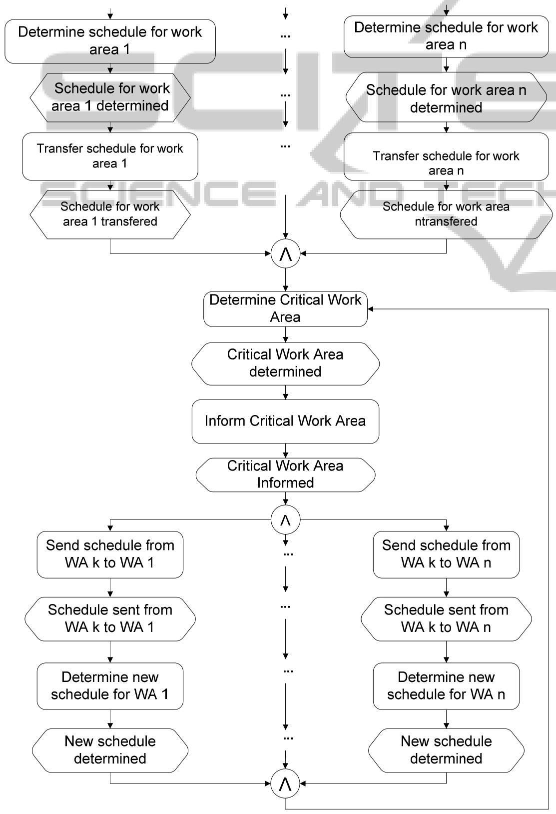

Figure 3: Functionality of the top- and base-layer (Part 2).

3.2 Identification of Services

Next, we are interested in deriving a concrete service model for the researched scenario. The basic idea for identifying services consists in determining for each function within an EPC or activity within an activity diagram, respectively whether it has to be implemented as an operation of a service or not. We use the following five criteria to make this decision:

1. **Degree of Automation (DA):** If no manual intervention is required between two consecutive functions then the two functions can be pooled together into one operation, otherwise, the two functions have to be represented by different operations.
2. **Atomicity (A):** When sub-functions of a certain function can be used within different business processes then it makes sense to implement the sub-functions using different operations. When the entire function can be re-used then a service might be implemented by orchestrating the services that correspond to the sub-functions.
3. **Modularity (M):** Functions are often carriers of an algorithm. When different algorithms are available to solve a problem, a situation-dependent selection of one algorithm is often beneficial. When such a selection is required, then an implementation of the function within a service is useful.
4. **Reusability (R):** When a function is applied in different business processes, then the function has to be implemented within a service.
5. **Number of Users (NU):** When a function is used by different humans, a certain degree of reusability is given. This function has to be implemented within a service.

Note that some of these criteria are also contained in (Kohlborn *et al.*, 2009 and Mönch and Zimmermann, 2009).

Applying these five criteria, we obtain a set of operations that can be grouped into different services. We differentiate between different types of services according to Kohlborn *et al.* (2009). Object-centric services are given by a set of operations that are used to access business objects. Task-centric services are formed by a set of operations that can be carried out without a specific business object. Hybrid services are somewhere between object- and task-centric services, because they consist of operations that are used to access data and perform tasks. Finally, process-centric services are used to support business processes using operations of other services by means of service composition. The first three service types correspond to basic operations that cannot be further decomposed. The resultant services are shown in Table 1. Here, we use the abbreviations PPC and PM for production planning and control and preventative maintenance, respectively.

Determining the necessary data for the job planning functionality is important. Therefore, we decompose the corresponding function into several sub-functions that are implemented as operations of

SCITEPRESS SCIENCE AND TECHNOLOGY PUBLICATIONS

a service that is related to data management for job planning. Several algorithms can be used to find appropriate parameters for the algorithms used for job planning. Therefore, this function is identified as an own operation of a job planning-related service. Because different job planning algorithms are possible, we also identify the function that determines job plans as an operation of the job planning service. The function that transfers job plans to work areas is only important in the context of work area scheduling. However, within the flow control, the job plans are available. Therefore, this function is not identified as an operation of the job planning service.

Table 1: Applying the criteria to the scheduling functions.

Criterion	DA	A	M	R	NU
Data Gathering Job Planning	0	X	0	X	None
Parameter Setting Job Planning	X	0	X	X	PPC
Calculate Job Plans	X	0	X	X	PPC
Transfer Job Plans to Work Areas	0	X	0	0	None
Data Gathering for Scheduling a Work Area	0	X	0	X	None
Parameter Setting Job Planning	X	0	X	X	PC
Determine Schedule for Work Area	X	0	X	X	PC, PM
Transfer Schedule to Top-Layer	0	X	0	0	None
Determine Critical Work Area	X	0	X	X	PPC
Inform Critical Work Area	0	X	0	0	None

The data management service for work area-related scheduling is justified in a similar manner as the data management service for job planning. Again, a refinement into several sub-functions is performed. Due to different possible algorithms for parameter setting within work area scheduling algorithms, we identify a corresponding parameter setting operation for the work area scheduling service. Different scheduling algorithms, e.g., decomposition- or local-search-based approaches, can be used to determine schedules for jobs in each single work area. Therefore, this function is represented by an operation of the work area-related scheduling service. Because the work area schedules are available within the flow control, we do not identify a separate operation to transfer schedules to the top-layer of the hierarchy. Several methods are available to determine a critical work area. Therefore, this function is identified as an operation of the job planning-related service. It is reasonable to inform the decision-making entity of a critical work area. Therefore, we include this functionality into the operation that determines the critical work area.

Table 2: Identified services for the scheduling process.

Service	Operation
Object-centric	
DataManagement-JobPlanning	ReadCapacityMS()/ ChangeCapacityMS()
	ReadJobsMS()/ AddJobsMS()
	ReadAggregatedRoutes-MS()/AddAggregated RoutesMS()
DataManagement-SchedulingWork-Area	ReadMachinesWA()/ AddMachinesWA()
	ReadJobsWA()/ AddJobsWA()
	ReadPartialRouteWA()/ AddPartialRoute()
hybrid	
JobPlanning	InitializeJP()
	SetParametersJP()
	ReadDataJP()
	DetermineJP()
	EvaluateJP()
	DetermineCriticalWA()
SchedulingWA	InitializeSWA()
	SetParameterSWA()
	ReadDataSWA()
	DetermineScheduleWA()
	EvaluateScheduleWA()
	IsWACritical()
process-centric	
Data Manage-mentScheduling	DetermineAggregatedCapa-cities()
AutomatedSche-dulingManufacturing System	StartProcess()

After the identification of operations, they have to be grouped into appropriate services as discussed before. We show the results of this second step in Table 2.

SCITEPRESS SCIENCE AND TECHNOLOGY PUBLICATIONS

Note that two process-centric services are described in Table 2. The first service is related to data management issues of an automated scheduling process. This service is based on operations of the DataManagementSchedulingWorkArea service. For example, the single machine capacities are determined using the ReadMachinesWA() operation. They are aggregated to capacities of certain work centers that are used within the job planning approach. The second process-centric service represents the automated hierarchical scheduling process. It contains only a single operation to launch this process. Note that the SchedulingWA service allows for the integration of problem specific scheduling approaches for different work areas.

4 IMPLEMENTATION OF THE PROTOTYPE

4.1 Implementation Issues

In this section, we will discuss our technology model. A prototype is designed and coded to check the feasibility of the proposed component. A client application implemented in the C# programming language provides a graphical user interface for calling the services and therefore, serves as a test driver. Web services, implemented in C#, are run on an ASP .NET development server. They implement the services described in Section 3.

We do not use a real MES for our prototype because we do not have access to an MES. Instead of that, the MES is simulated by a so called blackboard service, basically a SQL Server database. Within the prototype, we implement the AutomatedSchedulingManufacturingSystem service and the DataManagementScheduling service.

However, we are not interested in evaluating concrete scheduling algorithms for the scheduling scenario in this paper because this was done in previous research (cf. Mönch and Driessel, 2005 and Mönch *et al.*, 2006). Our main interest is related to the process flow. The architecture of the prototype is depicted in Figure 4.

4.2 Orchestration of the Services

An orchestration is required to implement the process-centric services. Orchestration of the services is performed using the BPEL engine of the Oracle BPEL process manager. The Oracle BPEL process manager runs on a J2EE application server. The decision to use Oracle BPEL is based on the fact that we used this tool for several previous prototypes.

Because web services are stateless, we compose them into a BPEL process that owns state variables that can be used to manage the state of the system. We can see from Figure 4 that the AutomatedSchedulingManufacturingSystem and the DataManagementScheduling services are implemented as BPEL processes.

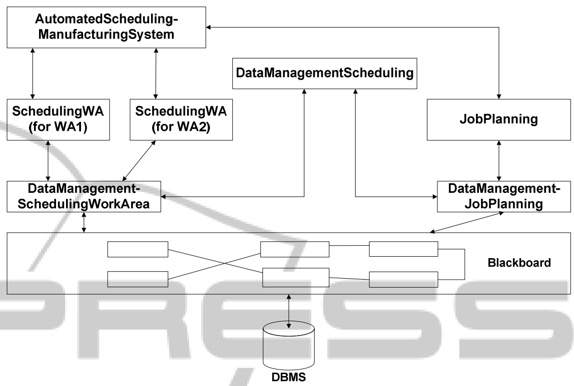

Figure 4: Architecture of the Prototype.

SOAP messages are used to exchange data between the different web services. This approach requires that appropriate XML data structures for routes, aggregated routes, job plans, and work area schedules are defined because SOAP is XML-based. Because we also have to write data, we use SOAP- and not REST-based web services.

4.3 Limitations of the Approach

There are some limitations of the proposed approach. Task objects, goals, and solution methods are the main building blocks of an enterprise task (cf. Ferstl and Sinz, 2006). Business processes work on task objects and change the attributes of these objects. Therefore, we can conclude that business processes are always related to persistency issues. Many object-centric services are the consequence (see Table 2). A tight coupling using joint master data is typical for scheduling and more generally for the production planning and control domain.

This may lead to undesirable side effects of services, such as problems with the global state consistency of the distributed application. The state of the art master data management in MESs has to be extended to fulfill the SOA requirements. We can see this requirement from Figure 4, where the blackboard service is used to represent the master data from the MES.

The second limitation is also related to data management in the service-based prototype. A certain

SCITEPRESS SCIENCE AND TECHNOLOGY PUBLICATIONS

number of XML-based SOAP messages have to be exchanged between the different operations of the services that form the scheduling component. A loss of performance might be the result of this situation.

5 CONCLUSIONS

In this paper, we described a scheduling component for complex manufacturing systems that is based on a hierarchical decomposition of the overall scheduling problem. We discussed the identification of appropriate services. The orchestration of these services is shown. The implementation of a prototype for such a component based on web services is also discussed.

There are some directions for future work. While it is possible to design and implement such a component, there is still much more effort required to integrate the resultant component into a real-world MES. A rigor assessment of the performance of the overall application, especially with respect to computing time, is also necessary.

Furthermore, it seems fruitful to combine software agents with service-oriented computing techniques as proposed by Huhns (2008). It is highly desirable to enrich the multi-agent-system FABMAS (cf. Mönch *et al.*, 2006) that implements a similar hierarchical scheduling approach as described in the present paper by web services. It is differentiated between decision-making and staff agents in FABMAS. Staff agents support the decision-making agents. It seems possible to replace the staff agents by web services as discussed in the present paper. The decision-making agents can be used to carry out a more sophisticated orchestration of these services.

ACKNOWLEDGEMENTS

The author would like to thank Daniel Kaiser for implementing the prototype and for interesting discussions on the topic of this paper.

REFERENCES

Brehm, N., Marx Gomez, J., 2007. Web service-based specification and implementation of functional components in federated ERP-systems. In *Proceedings Business Information Systems (BIS) 2007, LNCS 4439*, 133-146.

Ferstl, O., Sinz, E. J., 2006. *Grundlagen der Wirtschaftsinformatik*. Oldenbourg, München, Wien, 5th Edition.

Fordyce, K., Bixby, R., Burda, R., 2008. Technology that upsets the social order - a paradigm shift in assigning lots to tools in a wafer fabricator - the transition from rules to optimization. In *Proceedings of the 2008 Winter Simulation Conference*, 2277-2285.

Framinan, J., Ruiz, R., 2010. Architecture of manufacturing scheduling systems: literature review and an integrated proposal. *European Journal of Operational Research*, 295, 237-246.

Gaxiola, L., Ramirez, M. D. J., Jimenez, G., Molina, A., 2003. Proposal of holonic manufacturing execution systems based on web service technologies for Mexican SMEs. In *Proceedings HoloMAS 2003*, LNAI 2744, 156-166.

Huhns, M. H., 2008. Services must become more agent-like. *WIRTSCHAFTSINFORMATIK*, 50(1), 74-75.

Kellogg, D., Walczak, S. 2008. Nurse scheduling: from academia to implementation or not? *INTERFACES*, 37, 355-369.

Kohlborn, T., Korthaus, A., Chan, T., Rosemann, T., 2009. Identification and analysis of business and software services – a consolidated approach. *IEEE Transactions on Services Computing*, 2(1), 50-64.

McClellan, M., 1997. *Applying Manufacturing Execution Systems*. St. Lucie Press, Boca Raton.

Meyer, H., Fuchs, F., Thiel, K., 2009. *Manufacturing Execution Systems: Optimal Design, Planning, and Deployment*. McGraw-Hill.

Mönch, L., Driessel, R., 2005. A distributed shifting bottleneck heuristic for complex job shops. *Computers & Industrial Engineering*, 49, 673-680.

Mönch, L., Zimmermann, J., 2009. Providing production planning and control functionality by web services: state of the art and experiences with prototypes. In *Proceedings of the 5th Annual IEEE Conference on Automation Science and Engineering*, 495-500.

Mönch, L., Stehli, M., Zimmermann, J., Habenicht, I., 2006. The FABMAS multi-agent-system prototype for production control of wafer fabs: design, implementtation, and performance assessment. *Production Planning & Control*, 17(7), 701-716.

Pfund, M., Mason, S. J., Fowler, J. W., 2006. Semiconductor manufacturing scheduling and dispatching. Chapter 9 in *Handbook of Production Scheduling*, Herrmann, J.W. (ed.), Springer, New York, 213-241.

Qiu, R. G., 2007. A service-oriented integration framework for semiconductor manufacturing systems. *International Journal Manufacturing Technology and Management*, 10(2-3), 177-191.

Siedersleben, J., 2007. SOA revisited: Komponentenorientierung bei Systemlandschaften. *WIRTSCHAFTSINFORMATIK*, 49, 110-117.

Tarantilis, C. D., Kiranoudis, C. T., Theodoadoakopoulos, N. D., 2008. A web-based ERP system for business services and supply chain management: application to real-world process scheduling. *European Journal of Operational Research*, 187, 1310-1326.

Winkler, V., Buhl, H. U., 2007. Identifikation und Gestaltung von Services: Vorgehen und beispielhafte Anwendung im Finanzdienstleistungsbereich. *WIRTSCHAFTSINFORMATIK*, 49, 257-266.