Li LI, Fei QIAO

Tongji University

A Modular Simulation System for Semiconductor Manufacturing Scheduling

***Abstract**. A modular simulation system (MSS) for the scheduling of semiconductor wafer fabrication facilities (fabs) is discussed. Firstly, the general structure model (GSM) for modeling a semiconductor wafer fab, composed of a configurable definition layer, a physical layer, a process information layer and a scheduling layer, is proposed. Secondly, a data-based dynamic modeling method is given. Thirdly, the architecture of MSS, composed of data layer, module layer and simulation layer, is presented. Finally, a case study is used to demonstrate the effectiveness of MSS.*

***Streszczenie.** Zaprezentowano Isystem symulacji modularnej do planowania produkcji w przemyśle półprzewodnikowym. Bazuje się na warstwach: fizycznej, technologicznej i planowania. Zaproponowano też model dynamiczny. System składa się z trzech warstw: danych, modułu i symulacji. (**Modularny system symulacji do planowania w przemyśle półprzewodnikowym**).*

Keywords: Scheduling; Simulation; Semiconductor Manufacturing
Słowa kluczowe: planowanie, półprzewodniki, symulacja.

Introduction

Discrete event simulation technology, imitating a real production environment by modeling, has been applied to semiconductor manufacturing scheduling since 1980s. For example, Wein [1] built three kinds of HP24 simulation models to compare various release control strategies and scheduling rules and proved that the release control strategies may be more important to the performance of a semiconductor wafer fabrication facility (fab). Baudouin et al.[2] presented an information system architecture and a decision support tool to allow admissible scheduling solutions in a semiconductor device manufacturing fab. Thompson [3] developed a simulation-based finite capacity planning and scheduling software to allow human planners to make better decisions. This tool improved cycle time, throughput, and equipment utilization significantly without adding additional equipment and personnel.

Presently, discrete event simulation technology has been widely applied to predict the operational performance of a semiconductor wafer fab and assist its scheduling decisions. Existing fruitful research results can be classified into three main directions.

(1) Simulation systems performing scheduling behaviors

Some researchers developed various simulation systems to perform or assist scheduling behaviors of semiconductor wafer fabs. For example, Weigert et al. [4] developed a simulation-based scheduling system for a semiconductor Backend facility. The heuristic search strategies were adopted to optimize the operating sequences with consideration on concurrent minimization of mean cycle time, maximization of throughput and due date compliance. Zhang et al. [5] implemented a simulation platform based on extended object-oriented petri net for real-time scheduling of semiconductor wafer fabrication systems, and designed a dynamic bottleneck dispatching algorithm to detect bottlenecks in a timely way to make adaptive dispatching decisions according to the real-time conditions. Ramírez-Hernández et al. [6] presented and implemented the architecture of a preventive maintenance optimization software tool, based on Approximate Dynamic Programming algorithms, for the optimal scheduling of preventive maintenance tasks in semiconductor manufacturing operations.

Simulation as a tool to verify and improve the performance of scheduling algorithms

(2) Comparing to the achievements on the simulation-based scheduling system, the results of simulation as a tool to verify and improve the performance of scheduling algorithms are much more comprehensive. For example, Jeng and Tsai [7] applied simulation experiments to show their Match Due Date scheduling rule focused on the actual working mode of memory IC with the ability to reduce earliness and tardiness, make less number of setups, shorten the flow time and enhance the confirmed line item performance. Tang et al. [8] proposed a genetic algorithm and simulated annealing algorithm based scheduling method in semiconductor manufacturing lines and verified it with a Minifab simulation model. Chen and Wang [9] proposed a modified fluctuation smoothing rule incorporating a fuzzy-neural remaining cycle time estimator to improve scheduling performance in a semiconductor manufacturing factory, and evaluated its effectiveness by production simulation. Chou et al. [10] developed a simulated annealing algorithm with a probability matrix integrated with a greedy heuristic to solve the dynamic scheduling problem of semiconductor burn-in operations optimally in practical sizes. They proved that the proposed method could effectively and efficiently obtain optimal solutions for small size of problems and provide high-quality solutions efficiently for large size of problems by simulating.

(3) The evaluation and improvement of simulation-based scheduling methods

The importance of simulation-based scheduling methods has aroused many researchers' interests. They wanted to improve this method much better than ever before. For example, Li and Mason [11] investigated the potential advantages and drawbacks of using simulation-based scheduling in a semiconductor wafer fab. Results suggested the potential for simulation-based scheduling approaches to improve on-time delivery performance to customers in wafer fabs. Koyuncu et al. [12] augmented the validity of simulation models in the most economical way via incorporating dynamic data into the executing model, which then steered the measurement process for selective data update. Kim et al. [13] proposed a simplification method for accelerating simulation-based real-time scheduling in a semiconductor wafer fabrication facility. In the suggested real-time scheduling method, lot scheduling rules and batch scheduling rules were selected from sets of candidate rules based on information obtained from discrete event simulation. Since a rule combination that gave the best performance may vary according to the states of the fab, they suggested three techniques for accelerating rule comparison, too.

Although the simulation-based scheduling approaches in semiconductor manufacturing field have made big progresses, their common problem is lack of a standard scheduling simulation model. Consequently, quite a few of research results are applicable to a special production environment and difficult to be directly extended to other semiconductor wafer fabs to obtain better performance. Presently, there are some exploratory researches on standard simulation model. For instance, Ralph et al. [14] presented a prototype framework for scheduling of semiconductor manufacturing that divided the production system into Fabmodel, Process Route, Process Step, Tool Set and Operator Set. Rajesh and Sen [15] proposed a method to simplify the simulation models' complexity.

In this paper, we design a standard modular simulation system (MSS) for scheduling of semiconductor wafer fabs, characterized with modularization and decoupling the algorithms from the models according to scheduling characteristics of semiconductor wafer fabs. The remainder of this paper is organized as follows. A general structure model (GSM) of the scheduling of semiconductor wafer fabs is introduced in section 2. Section 3 presents a data-based dynamic simulation modeling method. Section 4 designs and develops MSS. Finally, we validate MSS with a case study in section 5. Section 6 contains conclusions and future works.

General structure model

In view of its physical structure, a semiconductor wafer fab is composed of dozens of work-centers. Each work-center includes multiple machines. Every machine has several or dozens of recipes to finish more than one operation of a job. An operation of a job can be finished by one or more machines and its processing time on different machines may be different.

In view of its process flow, a job will be processed as soon as it is released to a wafer fab. The machine processes a job according to an existing sequencing plan or directly selects a job in its queue according to some priority rule considering the real states of the wafer fab. A job's processing record is stored in the manufacturing execution system (MES). A machine needs a preventive maintenance at regular intervals.

In view of product definition, a job has its own process flow in a wafer fab. There are hundreds or thousands of jobs in a wafer fab at the same time. They may belong to different product editions. If the process flow of each product edition is stored in MES, the data storage will be huge. And the process flow of a product edition is difficult to be modified. In addition, a process flow is composed of hundreds of operation sets. An operation sets includes several operations. An operation is corresponding to several pairs of machine and recipe. An operation of one job may be the same with that of other jobs. So we only need storing these operations in MES and combining them into different process flows.

In view of scheduling of a semiconductor wafer fab, its objective is to reasonably utilize the machines and resources in the fab to achieve better operational performance and meet the requirements of the jobs' process flows and other constraints as well. In other word, the scheduling behavior is responsible for arranging right resources for an operation at right time. There are three main scheduling ways in a semiconductor wafer fab, i.e., static scheduling (usually called sequencing), dynamic dispatching and rescheduling. Static scheduling is to determine the processing order and time for specified operations before the real processing begins. Dynamic scheduling is to determine next processing route of a job or a processing priority of a job on a machine according to current states of a fab in a real-time style. Due to its uncertain manufacturing environment, an existing sequencing plan for a wafer fab may be not applicable to implement. Then the sequencing plan must be modified or regulated to be suitable to the real production environment. This behavior is called rescheduling.

Based on the analysis results as above, the scheduling of a semiconductor wafer fab can be abstracted as four parts, i.e., WIP-centered configurable definition, machine-concerned physical environment, process information recording production information, and scheduling describing detail algorithms. Then the general structure model of scheduling of a semiconductor wafer fab can be divided into static part and dynamic part (shown in Fig.1).

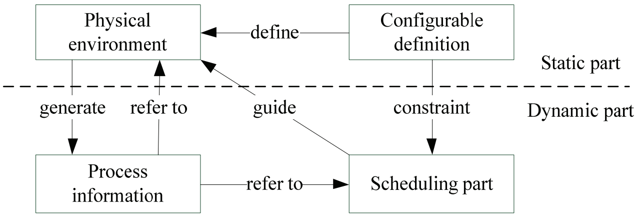

Fig. 1. Main part of General structure model (GSM)

The static part defines static information of a semiconductor wafer fab, including physical (machine-centered) and configurable (product-centered) definition. The dynamic part defines the information used during the production, including process information (e.g., release plan and sequencing plan) and scheduling related information (e.g., scheduling algorithms). Configuration definition regulates the production on the physical machines by defining the process of WIP. It also offers reference to the scheduling algorithms. Process information provides information (usually a sequencing plan) to machines to facilitate their production processes. On the contrary, the processing on a machine generates process information, too. In addition, process information acts as the input or feedback of scheduling algorithms.

.

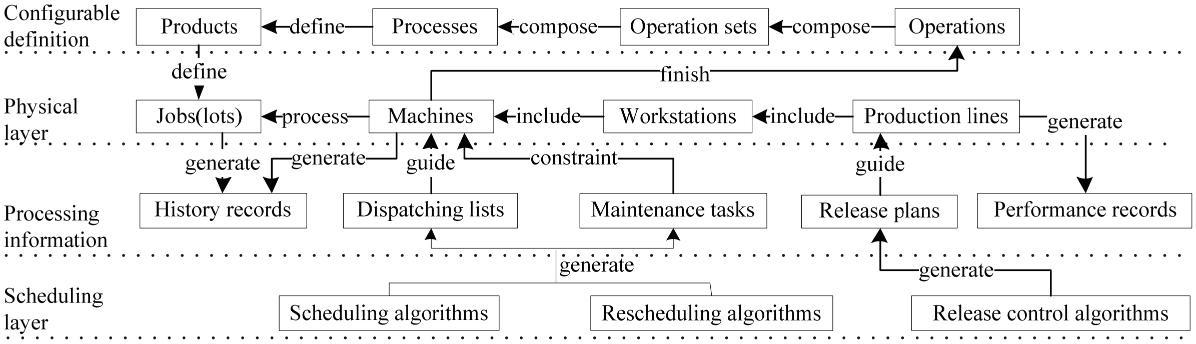

Fig. 2. General structure model (GSM) and its definition at each layer

Transferring those parts to its concept layer, a GSM of scheduling of semiconductor wafer fabs is also defined as four layers, i.e., configurable definition layer, physical layer, process information layer and scheduling layer. The components in each layer and the relations between these layers are shown in Fig.2.

The configurable definition layer manages process information of the products of a semiconductor wafer fab, such as products, processes, operation sets and operations. The minimum unit is the operation. Multi-operations compose an operation set, and multi-operation sets constitute a process. Every product has its own process specifications.

The physical layer records resource information of a semiconductor wafer fab, such as its layout and machines, the process abilities and state information of these machines, and state information of WIPs.

The process information layer records the data generated from the simulation processes, including input and output information. Input information contains release plans and machine maintenance plans. Output information includes dispatching lists, machine maintenance tasks, performance issues and the work log of the machines.

The scheduling layer contains some algorithms used by the simulation, including release control algorithms, scheduling algorithms and rescheduling algorithms. Release control algorithms generate the release plan during a special period. Scheduling algorithms, applied to the simulation running, generate the dispatching lists and the maintenance tasks for the machines during a special period. Rescheduling algorithms are used to modify the existing dispatching lists or generate new dispatching lists for a special period.

Data-based dynamic simulation modelling method

It is noticed that the differences between various semiconductor wafer fabs are their layouts, machines, processes and scheduling methods that can be transferred to data by some ways. If it is built dynamically by using these data, the simulation model is merely dependent on these data, which overcomes the deficiency of the strong pertinence. For example, Horn et al. [16] established a simulation-based interactive scheduling system, called “BackendPlanner”, for the backend facility of a semiconductor manufacturer under practical conditions. Its special characteristics included the automated model generation from existent production databases, a heuristic optimization component, a fast-simulator, and the efficient data coupling strategy allowed the re-scheduling anytime, e.g., after machine breakdowns or other unpredictable events and several additional analysis tools reported important parameters of the manufacturing process, such as machine utilization, throughput time or work in process. Werner et al. [17] developed a scheduling system for the backend of Infineon Technologies Dresden based on a discrete event simulation system and tested it in the real industrial environment. The simulation model was automatically generated from the databases of the manufacturer and was used for short term scheduling - from one shift up to one week. Sivakumar and Gupta[18] conceptualized, designed, and developed a discrete event simulation based "online near-real-time" dynamic multi-objective scheduling system to achieve Pareto optimal solutions in a complex manufacturing environment of semiconductor back-end. The system used a linear weighted aggregation optimization approach for multiple objectives and auto simulation model generation for online simulation. In addition, it enabled managers and senior planners to carry out "what now" analysis to make effective current decisions and "what if" analysis to plan for the future.

However, the former research results, focusing on a special semiconductor wafer fab, can only dynamically generate a simulation model. They can’t realize decoupling between scheduling algorithms and simulation models. As a result, the release control and scheduling algorithms used in a dynamic simulation model only fit to a special simulation model, which are difficult to be applied to other simulation models to obtain better performance. In addition, it is not easy to evaluate those algorithms’ performance under different production environments.

The above-mentioned less generalness of simulation models can be solved by using the proposed GSM in section 1.

Data Types for Dynamic Modelling

The process of dynamic modeling of the scheduling simulation model of semiconductor wafer fabs is to upload data, handle data, and finally organize data into a simulation model with specified structure. Each data unit belongs to some module in a layer of GSM. So the first job to analyze the require data types for dynamic modeling of a simulation-based scheduling model is to analyze the data requirements of each module in GSM.

Based on the definition on GSM, data types required by the simulation-based scheduling model are summarized into Table 1.

Table 1. Data types required by dynamic modeling

Layers	Modules	Require	Data types
Configurable definition layer	Product, process, operation set and operation	Yes	Product data
Physical layer	Job, machine, work centre and wafer fab	Yes	Physical resource data
Process information layer	History records, sequencing plan and performance records	No	
	Release plan and preventive maintenance	Yes	Production data
scheduling layer	Static scheduling, dynamic scheduling and on-line optimization (i.e., rescheduling)	Yes	Algorithm data

(1) Product data

Product data is corresponding to the configurable definition layer in GSM. Because every wafer fab has its own products’ definition, the product data is required without doubt for dynamic simulation modeling. Product data includes information related to products and relations between operation sets and machines. The latter defines the relations between physical layer and configurable layer in GSM.

(2) Physical resource data

Physical resource data is corresponding to the physical layer in GSM. It describes the physical layout of a semiconductor wafer fab. Because different wafer fabs have their own physical layout, the physical resource data is required without doubt for dynamic simulation modeling, too. Physical resource data is the main body of dynamic simulation model. It decides the numerical layout of a wafer fab in computer software. Traditional modeling methods directly set physical resource data into a special simulation model. If there are some changes (e.g., new machines addition) in the layout, the simulation model must be modified. However, data-based modeling is to build the numerical layout of a wafer fab dynamically according to physical resource data. Therefore, if there are changes in the physical layout of a wafer fab, no extra work is required during modeling process. The dynamic data-based modeling method is more flexible and general than traditional modeling methods.

(3) Process data

Process data, corresponding to the process information layer in GSM, includes the information required by production processes, e.g., maintenance tasks and release plans. It is the precondition to run the simulation. Process data doesn't include information about the historical records, sequencing plans and performance records that are generated during the simulation running process. So these data is unrequired during model uploading process.

(4) Algorithm data

Algorithm data, corresponding to the scheduling layer in GSM, records the scheduling algorithms applied to a wafer fab. The detailed scheduling algorithms are realized in the simulation model. Algorithm data records the information related to the algorithms, including algorithm names and parameters.

Once these four kinds of data are self-contained, the simulation model of a semiconductor wafer fab can be loaded dynamically successfully. The flexible superiorities of dynamic data-based simulation model comparing to traditional simulation model are as follows.

Firstly, these four data types are relatively independent and can be easily combined to build simulation models of different semiconductor wafer fabs. So the reusability of these data types is enhanced and the workload of simulation system development is seriously reduced.

Secondly, it is much easier to modify a simulation model. To modify a simulation model needs only the modification of the data required by generating the dynamic simulation model.

Dynamic Modelling Process

In view of traditional simulation modeling method, its structure (i.e., physical layer) is determined at the beginning of building the simulation model. Data is added to the physical layer. So if there are some changes in the physical layer, the data and model need corresponding modification with heavy workload. This kind of modeling process is called static modeling (shown in Fig.3). The data of static modeling will not be uploaded unless the simulation process starts.

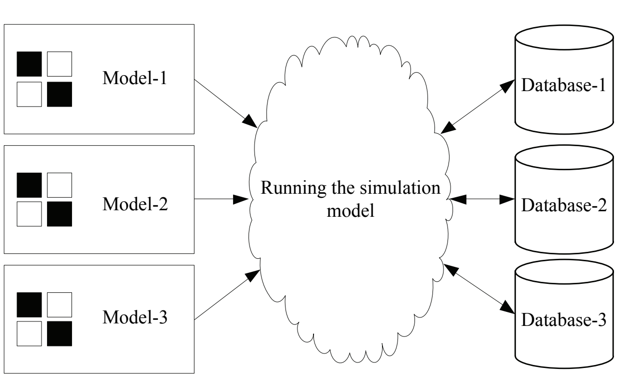

Fig. 3. Static modeling diagram

In view of dynamic data-based simulation modeling method, data takes an important role during the modeling process. On the contrary, the simulation model is an assistant. All of the changes of the simulation model are realized by changing the data. In other word, the simulation model of the same semiconductor wafer fab may be different due to the changes of its data. This kind of modeling process dependent on the data is called dynamic modeling (shown in Fig.4).

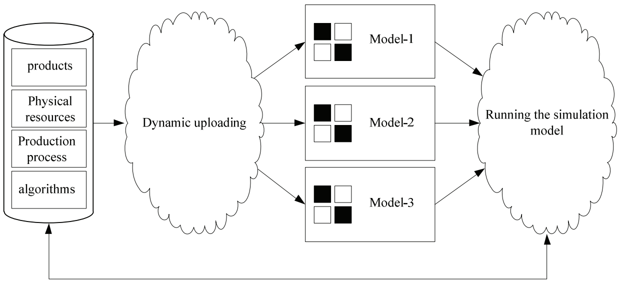

Fig. 4. Dynamic modeling diagram

The data-based dynamic simulation modeling flow is as follows: load the original data first, then deal with these data to standardize them into structural data required by dynamic modeling, and finally organize these structural data into a simulation model with a specified structure. These data will attribute to some modules of some layers of GSM ultimately.

Simulation-based modular scheduling system

Based on the proposed dynamic simulation modeling method in section 3, we design and develop a modular system for scheduling of semiconductor manufacturing (MSS). MSS integrates six semiconductor wafer fab models, including 4-inch wafer fab (BI4), 6-inch wafer fab (BI6), HP24Fab1, HP24Fab2, HP24Fab3 and Mini-FAB. Modular means that MSS integrates various kinds of scheduling in a wafer fab, such as release control, sequencing, dynamic dispatching and rescheduling methods. Each scheduling method is encapsulated as a module. The modules belonging to the same kind of scheduling can be replaced each other.

The hardware environment of MSS is Pentium IV 1.5 CPU, 256M memory and 10G disk driver or more. The software environment includes Windows 2000 or XP operation system, .NET Framework 2.0, Microsoft Office 2000 or higher edition, Graphics.Server.NET.v1.1 and simulation platform eM-Plant.

The architecture of MSS is shown in Fig.5. There are three layers in MSS, i.e., data layer, software layer and simulation layer.

(1) Data layer

Data layer, realized with an ACCESS database, storages the data of all semiconductor wafer fab models in MSS. These data are the base to build the basic model in software layer. There are communications between data layer and software layer. There are communications between data layer and simulation layer, too. These communications are enabled through corresponding data interfaces to facilitate free transfer between different semiconductor wafer fab models.

The design principle of data layer is to guarantee identical data structures of six semiconductor wafer fab models. The standard of the identity is the data types in Table 1. To meet the requirements of GSM on data, we build following data-entity tables in the ACCESS database.

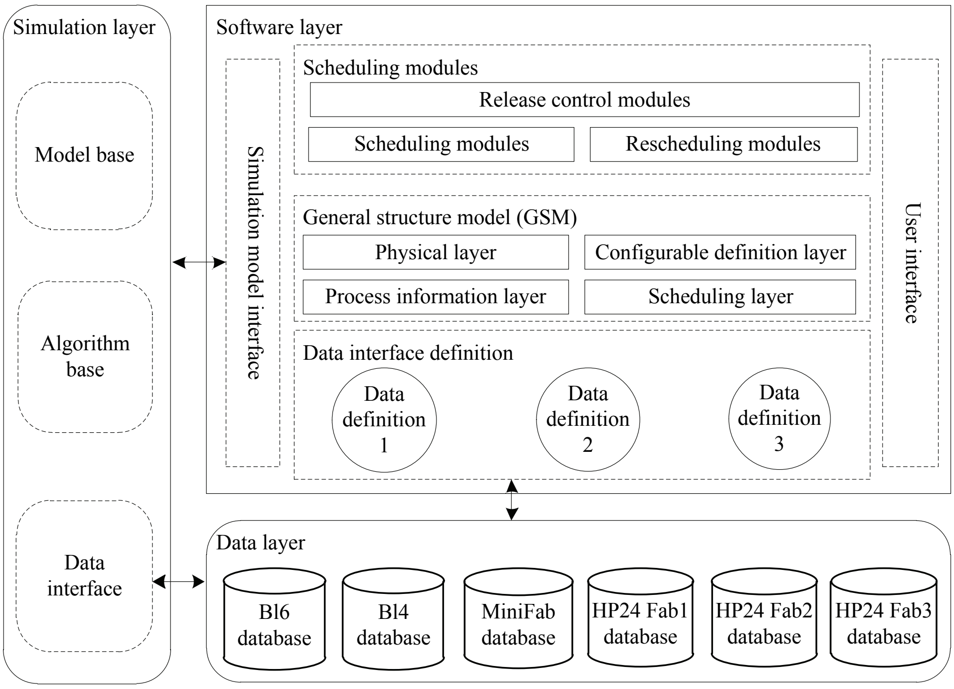

Fig. 5. Overall design of modular simulation system for scheduling of semiconductor manufacturing (MSS)

- Order table: the attributes include customer, priority, number, due date and etc. The order table is the base of release plans.
- Release table: the attributes include release number, release date, belonging products and etc.
- Maintenance task table: the attributes include the latest start time, the earliest start time, duration time and etc.
- Product table: the attributes include the definition of process flow, the processing time of operation, average cycle time of product, average move step and etc.
- Work-center and machine table: the attributes include the constitution of work-center, machine state, machine type, scheduling rule used by machine, processing history record and etc.
- WIP table: the attributes include WIP state, machine queuing, state start time, state finish time and etc.
- Dispatching list table: the attributes include operation start time, operation finish time, job number, machine number and etc.
- Performance table: include all the performance records in a wafer fab.
- Scheduling algorithm table: the attributes include the name of scheduling algorithm and the relations between algorithms and machines.

The relations between above-mentioned data-entity tables and data types in Table 1 are shown in Table 2.

Table 2. Relations between data-entity tables and data types

Data types	Data-entity tables
Product data	Product table
Physical resource data	Work-centre & machine table, WIP table
Production data	Release table, maintenance task table, dispatching list table, performance table
Algorithm data	Scheduling algorithm table

(2) Software layer

Software layer is the core of MSS. It is used to model a semiconductor wafer fab in a standard way, manage MSS's modules configuration, and provide a user management interface. Through specified data interface, software layer can read data from data layer. Then it configures the modules in it and sets the parameters of modules. Finally, it can call simulation layer to realize the simulating process of semiconductor wafer fab models.

Software layer includes five components, i.e., data interface definition, general structure model, scheduling modules, simulation interface and user interface. In addition, we add some functions to make statistics on system data, generate their diagrams and files, and manage the system parameters.

- Data interface definition

Data interface definition is to read data from data layer to software layer and organize these data according to the data types in GSM. Data layer stores data from six semiconductor wafer fab models. Although their data are different, the structures of their databases are the same.

So we can only develop one data interface to read either model. If there are other databases with different structures, we only need to define corresponding data interfaces to integrate these heterogeneous databases easily.

- General structure model

General structure model organizes the data read from data layer into data types described in Table 1 to facilitate scheduling modules to call them quickly.

- Scheduling modules

The scheduling algorithms in MPSS are realized as multiple modules. A scheduling module obtains the algorithm configuration information from user interface, then configures the algorithm's parameter and implements the computation process, finally sends the computation results to the user through the user interface.

- Simulation interface

Simulation interface is the communication interface between software layer and simulation layer. Its functions include: the open, close and save of simulation models, simulation control (i.e., simulation start, pause and reset), and implementation of simulation inner script language.

- User interface

User interface is the interactive interface for user operating software layer. The design principle of use interface is full-open, i.e., the panels for setting parameters are all on the main frame and reducing the number of menus as less as possible.

(3) Simulation layer

The simulation layer is to simulate a current semiconductor wafer fab model. It takes eM-Plant 7.5 as its platform, including a model base, an algorithm base and a data interface.

- Model base

There are six semiconductor wafer fab simulation models in the model base. These models are generated dynamically according to the method introduced in section 3. So the structure, control programs and data interface of these models are identical.

- Algorithm base

There are four parts in algorithm base: model uploading algorithm, simulation model control algorithm, scheduling rules and statistical algorithm. These algorithms are applicable to all simulation models in the model base.

- Data interface

Data interface is used to interact with database to read original data from data layer and write simulation data back to the database.

Case study

MSS can be used as a platform to test the performance of different scheduling algorithms and the proper mix level of scheduling algorithms. Thus, it can be considered as a reference tool for the production decisions.

The workflow of MSS includes uploading simulation model, generating release plan, generating sequencing plan, dispatching and rescheduling (shown in Fig.6).

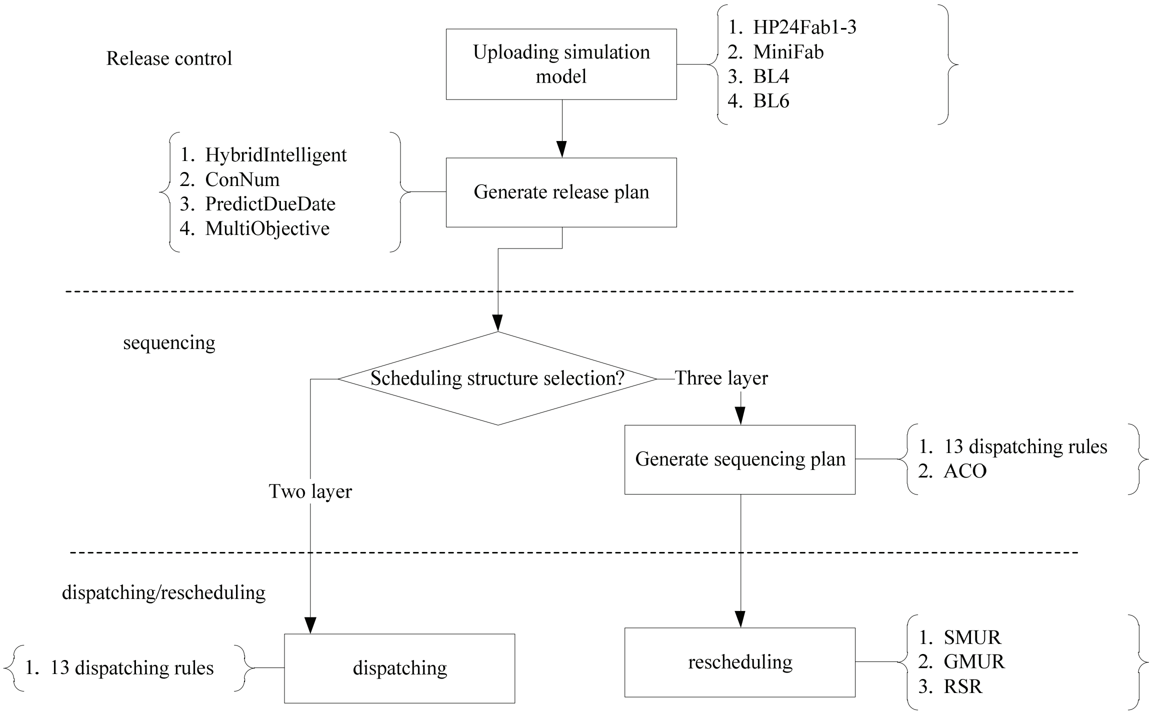

Fig. 6. MSS workflow and its main modules

(1) Uploading simulation model

The uploading simulation model process is to transfer data from data layer to software layer. Due to huge amount data in semiconductor manufacturing, this process may spend longer time. The uploading time for six simulation models is shown in Table 3.

Table 3. Uploading time of six simulation models

Simulation model	Uploading time (s)
MiniFab	4.1 ~ 4.3
HP24Fab1	23.1 ~ 23.3
HP24Fab1	22.9 ~ 23.2
HP24Fab1	20.3 ~ 20.5
Bl4	110.5 ~ 111.1
Bl6	168.7 ~ 169.6

(2) Release control modules

There are four release control modules, i.e., ConNum (the number released to the fab per day is the same), HybridIntelligent (the release plan is generated by an immune algorithm), PredictDueDate (the jobs released to the fab according to the urgent level of their expected due dates) and MultiObjective. Taking Minifab as an example, the performance of each release control modules are shown in Table 4.

(3) Scheduling modules

The scheduling algorithms in MSS include FIFO, EDD, EODD, LPT, SPT, CR, FSVCT, LS, FIFO+, SRPT, SRPT+, SRPT++, FSVL and DBR. The release period is set to 4 months. We select the data in the middle 2 months to implement analysis work. The judge criterion is set to the product of the on-time delivery and the throughput. The former 10 matching of the release control algorithms and the scheduling algorithms are shown in Table 5 [19]. Obviously, the matching of HybridIntelligent and DBR is the best decision.

Table 4. Performance statistics of release control modules

Release control	Cycle time (min)	Cycle time variance (min)	On-time delivery rate (%)	Throughput (lot/day)	MOV (lot)	WIP (lot)
HybridIntell-igent	4780.38	3009.65	74%	15.17	99	57.17
ConNum	6728.39	3267.84	74%	15.47	100.22	76.33
PredictDue-Date	4902.45	3500.41	97%	15.27	100.00	59.47
MultiObjec-tive	6580.04	2359.95	50%	15.67	100.9	72.73
Average	5747.82	3034.46	74%	15.40	100.03	66.43

Table 5. Good matching sequence

Release control	Scheduling algorithm	On-time Delivery (%)	Throughput (lot/day)	Judge criterion
HybridIntelligent	DBR	75.86	4.35	3.29991
HybridIntelligent	SPT	77.17	4.23	3.264291
ConNum	DBR	75.88	4.28	3.247664
PredictDueDate	DBR	80.08	4.02	3.219216
HybridIntelligent	FIFO	75.69	4.25	3.216825
HybridIntelligent	SRPT++	75.69	4.25	3.216825
PredictDueDate	EODD	78.05	4.1	3.20005
HybridIntelligent	EODD	75.29	4.25	3.199825
HybridIntelligent	SRPT+	75.29	4.25	3.199825
PredictDueDate	FSVCT	75.59	4.23	3.197457

Conclusions

MSS adopts a data-based dynamic simulation modeling method and a module-based decouple of the algorithms from the simulation models to enhance the simulations' efficiency and simulation models' reuse. It offers a better impersonal evaluation platform for further research on the scheduling algorithms.

Our future work is to add some intelligent planning and scheduling modules to SMPSS and optimize the data layer by using MYSQL, ORACLE or SQL SERVER instead of ACCESS to enhance the response speed of the system. Then it can meet the requirements of the factories better than ever.

Acknowledgment

The support of the National Natural Science Foundation of China under grant No.50905129 and No.61034004, and the Program for Scientific Innovation of Shanghai under grant No.09DZ1120600 and No.10DZ1120100 is gratefully acknowledged.

REFERENCES

[1] L. M. Wein, “Scheduling Semiconductor Wafer Fabrication”, IEEE Transactions on Semiconductor Manufacturing, vol.1, No.3, pp115-130, August 1988.

[2] M. Baudouin, C. Ruberti, J. Arekion, etc., “Decision Support System Based on a Factory Wide Information Integrated System and Discrete Event Simulation to Help Solve Scheduling Problems in a Semiconductor Manufacturing Environment”, Proc. of the IEEE Symposium on Emerging Technologies & Factory Automation, Paris, France, pp437-445, October 10-13, 1995.

[3] M. Thompson, “Simulation-Based Scheduling: Meeting the Semiconductor Wafer Fabrication Challenge”, IIE Solutions, vol.28, No.5, pp30-34, May 1996.

[4] G. Weigert, A. Klemmt, S. Horn, “Design and Validation of Heuristic Algorithms for Simulation-Based Scheduling of a Semiconductor Backend Facility”, International Journal of Production Research, vol.47, No.8, pp2165-2184, January 2009.

[5] H. Zhang, Z. Jiang, C. Guo, “Simulation-Based Optimization of Dispatching Rules for Semiconductor Wafer Fabrication System Scheduling by the Response Surface Methodology”, International Journal of Advanced Manufacturing Technology, vol.41, No.1-2, pp110-121, March 2009.

[6] J. A. Ramírez-Hernández, J. Crabtree, X. Yao, etc., “Optimal Preventive Maintenance Scheduling in Semiconductor Manufacturing Systems: Software Tool and Simulation Case Studies”, IEEE Transactions on Semiconductor Manufacturing, vol.23, No.3, pp477-489, August 2010.

[7] W. D. Jeng, M. S. Tsai, “Scheduling Semiconductor Final Testing a DBR Based Simulation Model”, Proc. of the 40th International Conference on Computers and Industrial Engineering: Soft Computing Techniques for Advanced Manufacturing and Service Systems, Awaji, Japan, July 25-28, 2010.

[8] C. H. Tang, Y. L. Qian, J. Zhu, etc., “A Scheduling Method in Semiconductor Manufacturing Lines Based on Genetic Algorithm and Simulated Annealing Algorithm”, Proc. of the 2010 International Conference on Information, Networking and Automation, Kunming, China, pp1429-1432, October 17-19, 2010.

[9] T. Chen, Y. C. Wang, “A Nonlinear Scheduling Rule Incorporating Fuzzy-Neural Remaining Cycle Time Estimator for Scheduling a Semiconductor Manufacturing Factory-A Simulation Study”, International Journal of Advanced Manufacturing Technology, vol.45, No.1-2, pp110-121, November 2009.

[10] F. D. Chou, H. M. Wang, P. C. Chang, “A Simulated Annealing Approach with Probability Matrix for Semiconductor Dynamic Scheduling Problem”, Expert Systems with Applications, vol.35, No.4, pp1889-1898, November 2008.

[11] W. Li, S. J. Mason, “Comparison of Simulation-Based Schedule Generation Methodologies for Semiconductor Manufacturing”, Proc. of the IIE Annual Conference and Expo 2007, Nashville, TN, United states, pp1387-1392, May 19-23, 2007.

[12] N. Koyuncu, S. Lee, K. K. Vasudevan, etc., “DDDAS-Based Multi-Fidelity Simulation for Online Preventive Maintenance Scheduling in Semiconductor Supply Chain”, Proc. of the 2007 Winter Simulation Conference, Washington, DC, United states, pp1915-1923, December 9-12, 2007.

[13] Y. D. Kim, S. O. Shim, B. Choi, etc., “Simplification Methods for Accelerating Simulation-Based Real-Time Scheduling in a Semiconductor Wafer Fabrication Facility”, IEEE Transactions on Semiconductor Manufacturing, vol.16, No.2, pp290-298, May 2003.

[14] M. Ralph, A. Christos, F. M. Leon, “Automatic Generation of Simulation Models for Semiconductor Manufacturing”, Proc. of the 2007 Winter Simulation Conference, Washington DC, USA, pp648-657, December 9-12, 2007.

[15] P. Rajesh, A. P. Sen, “Simplification Strategies for Simulation Models of Semiconductor Facilities”, Manufacturing Technology Management, vol.15, No.7, pp618-625, July 2004.

[16] S. Horn, G. Weigert, P. Schönig, etc., “Application of Simulation-Based Scheduling in a Semiconductor Backend Facility”, Proc. of the 1st Electronics System integration Technology Conference, Dresden, Saxony, Germany, pp1122-1126, September 5-7, 2006.

[17] S. Werner, S. Horn, G. Weigert, etc., “Simulation Based Scheduling System in a Semiconductor Backend Facility”, Proc.of the 2006 Winter Simulation Conference, Monterey, CA, United states, pp1741-1748, December 3-6, 2006.

[18] A. I. Sivakumar, A. K. Gupta, “Online Multiobjective Pareto Optimal Dynamic Scheduling of Semiconductor Back-End Using Conjunctive Simulated Scheduling”, IEEE Transactions on Electronics Packaging Manufacturing, vol.29, No.2, pp99-109, April 2006.

[19] L. Li, F. Qiao, “Design and Validation of a Simulation-Based Modular Planning and Scheduling System of Semiconductor Fabrication Facilities”, Applied Mechanics and Materials, vol.48-49, pp378-381, Jan. 2011.

***Authors**: prof. dr Li Li, School of Electronics and Information Engineering, Tongji University, Shanghai, China, E-mail: lilia@tongji.edu.cn.*