Development of the Order Fulfillment Process in the Foundry Fab by Applying Distributed Multi-Agents on a Generic Message-Passing Platform

Chih-Yuan Yu and Han-Pang Huang, *Member, IEEE*

***Abstract*—For a semiconductor foundry fab, the satisfaction of customers is critical. In order to provide a better service for customers, the modeling and simulation of the order fulfillment process (OFP) is important for the foundry fab since supervisors can use them to find the bottleneck in the process. The scope of this paper is to (1) model the OFP of a foundry fab in the distributed environment using multi-agents; (2) provide the functionalities for each application (agent) in the OFP; and (3) provide a generic message-passing platform (GMPP) for the distributed environment. The OFP can be viewed as the enterprise-scale integration of applications. It is decomposed into four subprocesses: order management process, planning process, manufacturing execution system (MES), and event monitoring. Each application in the subprocess is regarded as an agent and performs its task based on its knowledge constrained by its objectives. Agents are located in the distributed environment in order to possess the properties of a distributed system. The message types in the GMPP are classified into application-to-application, application-to-person and person-to-application. The enabling communication protocols, such as COM+ event, NET-NET and a gateway, which is the protocol converter between applications and users, will be addressed. The entire OFP is built on the GMPP. Some useful information for decision support systems is shown in the simulation results.**

***Index Terms*—Distributed environment, event solution set (ESS), foundry fab, message passing, multi-agents, order fulfillment process (OFP).**

I. Introduction

WITH THE GROWTH of the Internet, the legacy system cannot fulfill all the functions required by an enterprise. The integration of application islands is getting more important. Moreover, it is impossible to renew all existed applications in the enterprise to communicate with a new application. To solve this problem, a distributed environment, which can provide a transparent, distributed, loading balancing, fault tolerant, extendable, scalable and secure integration environment, is developed. Many researches [4], [6] discussed about the properties of the distributed environment. The distributed environment has two types. One is distributed computing performed by several processors and the other is distributed applications executing their own tasks. The distributed environment mentioned in this paper is the latter one. The applications executed in the distributed environment are divided into the shop floor level, manufacturing execution system (MES) level, and enterprise level, as shown in Fig. 1. DCOM and CORBA [5] enabling technologies are widely adopted to construct manufacturing automation systems.

For a semiconductor foundry fab, the satisfaction of the customers is the key to success. Recently, the way to reduce the cycle time in a high-yield rate fab and to provide the customer real-time information of orders are widely discussed. Taiwan Semiconductor Manufacturing Company (TSMC) proposed the concept of "Virtual Fab" and provides customers five modules, i.e., TSMC eFoundry, TSMC-Direct, TSMC-Online, TSMC-YES and TSMC-iLaView. Those modules provide not only the real-time information of orders but also the technology sharing. United Microelectronics Company (UMC) developed a similar concept called virtual foundry consultant that can assist their customers to design and obtain the status of lots. Su *et al.* [14] proposed an enabling framework for virtual fab and concluded that the virtual fab is the critical aspect for achieving competitiveness in the semiconductor industry. Lin *et al.* [8] used Queueing Colored Petri net (QCPN) to model the virtual fab. Tang [16] introduced the virtual production lines to configure a large semiconductor system into many production lines so that the line balance and efficiency were ensured.

In short, service, scalability, transparency, and extensibility are the keys for a foundry fab to construct its manufacturing system. The scope of this paper is to: (1) model the order fulfillment process (OFP) of a foundry fab in a distributed environment using multi-agents; (2) provide the functionalities for each application (agent); and (3) provide a generic message-passing platform (GMMP) for distributed environment.

The first focus of this paper is to develop the OFP in a foundry fab. The OFP includes four subprocesses, i.e., order management process, planning process, MES and event monitoring. These four subprocesses describe the activities triggered by receiving an order in the foundry fab. Agents are constructed in each subprocess. Each agent has its goals to achieve and its functions to execute. Hence, the functionality of each agent is also the emphasis of this paper.

The synchronization is a big issue in the distributed communication. Some communication requires synchronization while others not. For example, the customer places an order and the order management server should check the inventory immediately. The order acceptance and the order placement should be synchronized. Although the order should trigger the OFP to

Manuscript received February 15, 2001; revised October 2, 2001. Recommended by Guest Editor H.-P. Huang. This work supported in part by the National Science Council in Taiwan under Grant NSC 90-2212-E-002-222.

The authors are with the Robotics Laboratory, Department of Mechanical Engineering, National Taiwan University, Taipei, 10660, Taiwan, R.O.C. (e-mail: hphuang@w3.me.ntu.edu.tw).

Publisher Item Identifier S 1083-4435(01)10744-1.

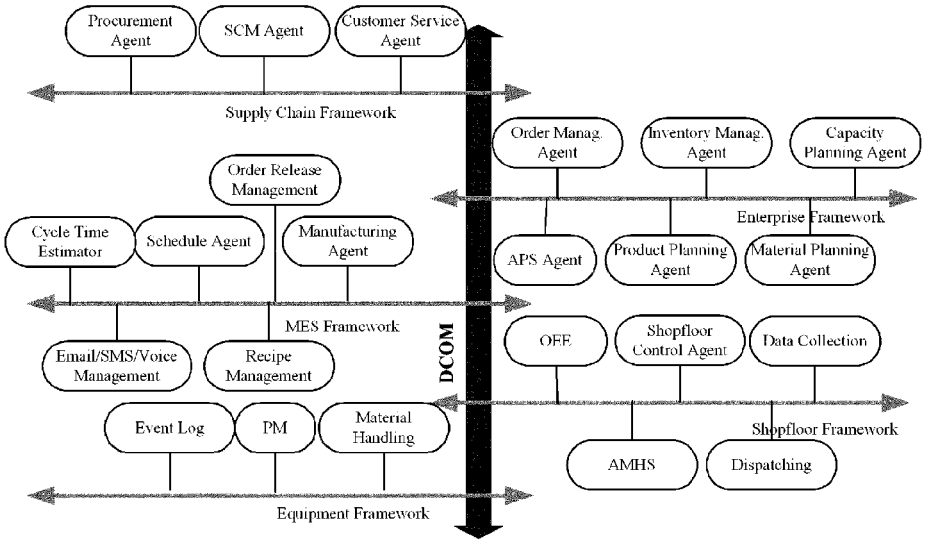


Fig. 1. Distributed environment for enterprise automation.

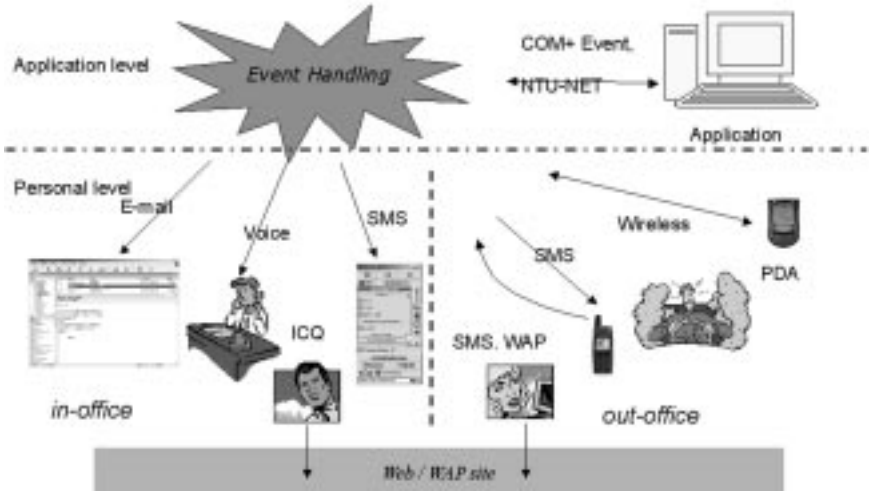


Fig. 2. The illustration of the message passing.

commit the order as soon as possible, the complete process can be regarded as an asynchronous one. No matter the communication is synchronized or not, there always exists many messages passing across the applications in the distributed environment.

Fabian *et al.* [3] proposed a message-passing structure for an flexible manufacturing system (FMS) controller. Since the structure is tightly coupled, the message should be designed based on the design time. It is less scalable for system extension. Object Management Group (OMG) has published the event service for its common objects. Ho *et al.* [4] especially addressed the loading balance and fault tolerance under the CORBA environment. However, there are some noticeable weaknesses [13]. For example, the efficiency of communication will decrease when the number of publishers and subscribers increase.

This paper also proposes a GMPP for distributed environment. The advantages of the GMPP are: (1) it provides the way for communication between applications and users; (2) the communication is scalable, transparent based on COM+ loosely coupled events; (3) the communication is secure, prioritized as applying NTU-NET [7]; (4) the application can notify and alarm users when errors occur; (5) users can monitor and control the application through the Web/WAP site; and (6) the error exception and the event solution set (ESS) is built in the gateway that transfers messages between applications and users.

The organization of this paper is as follows. The GMMP for distributed agents is proposed in Section II. In Section III, the multi-agent technology for constructing the OFP is described. Modeling of the OFP and the functionalities of each agent in the OFP are addressed in Section IV. The results are shown in Section V and conclusions are made in Section VI.

II. GMPP

A. Platform Overview

The GMPP is developed for communication between applications and users in a different and distributed environment. The platform is generic so that it is suitable for all kinds of applications and terminals that people hold. Fig. 2 shows the communication in the manufacturing automation system. The information is delivered to an application to inform or trigger an-

other process. Since the distributed applications are located in different computers, the message passing in the application level in Fig. 2 requires security, transparency, speed and reliability of the communication.

On the other hand, if an error occurs and the automation mechanism cannot handle this situation, an alarm or notification should be sent to the corresponding manager or engineer. Besides, the supervisor can monitor the shop floor situation by receiving the information sent from shop floor computers. At this point, the message passing in the person level includes in-office and out-of-office. The application can send the alarm message to the receiver by calling, voice broadcasting, emailing to his/her mailbox, sending a message to his/her ICQ number and sending a short message to his/her cellular phone.

After receiving the message from the application, the receiver might have to do some actions. For example, if an error occurs in the machine, engineers receive the error notification and handle this problem. If the problem is remotely solvable, such as software error and further notification to the manager, he can make the decision on the web site if he is in the office, or on the WAP site through his cellular phone if he is out of the office.

B. Functionality

The GMPP consists of three types of message passing: application-to-application (ATA), application-to-person (ATP) and person-to-application (PTA).

1) ATA: The "application" is defined as a standalone and executable entity in the distributed environment. It is also viewed as an agent in this paper. An application may communicate with other applications to share information or be notified as an event. In general, a distributed environment includes Internet. The application has to send the message across the WAN although most of the messages are routed in the LAN. As a result, the communication protocols, such as COM+ and NTU-NET, are suitable for ATA message passing.

2) ATP: In an automation process, all jobs are handled by applications. Computer integrated manufacturing (CIM) is the concept of manufacturing with the aid of computers. Unfortunately, some decisions should be made by persons who are not in front of the computer. The application might have to inform the supervisor to make decision or alarm engineers if errors occur in the shop floor. Engineers/supervisors can also monitor the shop floor situation when the application sends real-time information. Short message service (SMS), email, ICQ and voice are used as the message protocol for message reporting. Email and ICQ are the in-office message passing, SMS is the out-office message passing and voice is for both. People can receive voice message through telephone, SMS through cellular phone and email and ICQ through computer.

3) PTA: Compared to ATP, PTA provides a way for engineers/supervisors to respond to the notification, to control the in-office applications, or to make decisions as receiving a message. The following terminals are frequently used: cellular phones and personal digital assistants (PDA), which can be connected to WAP sites, personal computers, which can be connected to Web sites. As a result, htypertext transfer protocol (HTTP) is used to connect WAP and Web sites.

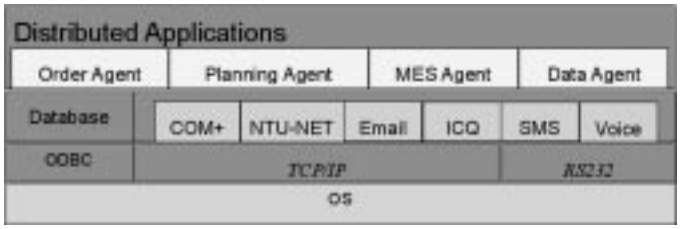


Fig. 3. The protocol hierarchy for distributed environment.

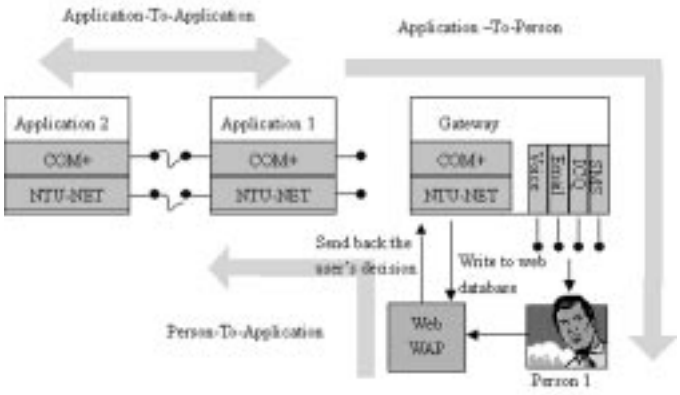


Fig. 4. The system architecture of the GMPP.

C. Communication Protocol

The communication protocols used in the GMPP are NTU-NET [7], COM+ [10], Email, ICQ, SMS [20], voice, and HTTP. NTU-NET and COM+ Event are used for ATA. For ATP and PTA, applications use Email, ICQ, SMS and Voice to notify engineers/supervisors. On the other hand, engineers/supervisors can control, or make decision through the Web/WAP site that is built on HTTP protocol.

D. Architecture and Implementation of GMPP

1) System Architecture: In the protocol hierarchy shown in Fig. 3, COM+, NTU-NET, Email, and ICQ encapsulate TCP/IP, while SMS and Voice use RS232 to send the message. The applications can include the COM+ event or NTU-NET to communicate with each other. As to the ATP, a protocol transfer gateway is constructed, which can be invoked by other applications via COM+ or NTU-NET and sends the message to engineers/supervisors via SMS, Email, ICQ and Voice. The role of gateway is shown in Fig. 4. The Web/WAP site provides the entrance for engineers/supervisors to take action then notify the corresponding application to respond.

2) Implementation of the Protocol Transfer Gateway: The protocol transfer agent provides a standard way for applications to send the message to engineers/supervisors through COM+ or NTU-NET. Messages sent to engineers/supervisors are classified into two types. One is without acknowledgment especially for ATP. The message is simply sent and the gateway does not detect whether engineers/supervisors receive it or not. The other needs the person to go to the Web/WAP site to respond messages. This combines ATP and PTA.

The gateway is also an application, which combines the email control, SMS control and serial communication control

(RS232). The SMS control is also used to send the message to the ICQ number. Besides, the SMS control connects the computer and the cellular phone by a cable through RS232. It is a simple SMS sender compared to the facility in the telecommunication company.

In the second type of the message in the gateway, two situations should be highlighted. What kinds of decision should a receiver make? How to control the exception of sending message? To solve the first problem, the ESS is proposed. The ESS is the set of solution for the corresponding event. The application that wants to get the response from a person should provide the solution set when firing the event. When the gateway receives the sending request from the application, it routes the message to person via the protocol, which the application assigns and then writes the solution set to the database. The receiver gets the message and then logs in the Web/WAP site. He/she will see and choose the corresponding solution set. After the receiver makes decision, it is sent back to the application.

The exception of message sending occurs when a person does not receive or respond to the message after resending several times in a durable period. The gateway should pass the resending and return error message to the original sender, which assigns the resending count and the durable period when invoking the message.

III. Multi-Agents in the OFP

A. Using Multi Agents Technique

Intelligent agents are a new category of information society tools. The characteristics of intelligent agents can be classified into internal and external properties [1]. Internal properties are those internal abilities of the agent, such as autonomy, learning, proactivity, and reactivity. External properties are those abilities to interact with the environment and agents in the society, such as coordination, collaboration, and communication.

In other words, an agent is an active object, which possesses certain abilities to perform task and communicate with other agents to complete the goal of the system [9]. An agent has an internal behavior model, a functional component consisting of procedures/heuristics/strategies and a protocol for interacting with other agents [12]. Agents in the same environment form a society to achieve the global goals. Hence, multi-agents technology is widely adopted in manufacturing systems [12], supply chain coordination [9], simulation [15] and shop floor control. The OFP can be viewed as the enterprise-scale integration of applications. Each application in the OFP is regarded as an agent and performs its task based on its knowledge constrained by its objectives. Besides, the OFP requires the coordination and collaboration of agents in the enterprise to achieve the goals. Therefore, they can communicate with each other on the GMMP discussed in Section II.

B. Agent Communication

An agent should communicate with others. One may invocate the other's procedure. The invoking agent passes parameters through function to inform another agent in order to accomplish its goal. Then the return values tell the invoker the status of the function call. The methods for agent communication can be classified into three types: direct message passing, blackboard discussion, and mediator [1].

GMPP is the extension of the direct message passing. The COM+ event mentioned in Section II is used to communicate between agents while the COM+ event is not the direct message passing. The mechanism of COM+ event is loosely coupled between senders and receivers. The coordination between agents is the set of all events registered in the operating system. The sender just fires a certain event and the receivers (the numbers of receivers may be larger than one) get the event and do the corresponding action. Basically, the message is not transferred within one sender and one receiver. Any agent, which is interested in getting the event, can subscribe the event in the operating system. The routing of events is important in the mediator and is handled by the operating system. The communication method used in this paper is good for real-time communication since the blackboard discussion is time consuming while the mediator is difficult to implement.

IV. Modeling of the OFP

The OFP is modeled as a society of agents. The modeling approach is based on the object-oriented system analysis and design. The entire OFP is divided into four subprocesses: order management process, planning process, manufacturing process and event monitoring, as shown in Fig. 5. Each subprocess contains its own agent. Sequence diagrams of some agents are shown in Fig. 6.

The following sections describe the functionalities of each agent in the subprocesses and the implementation of the GMPP on the multi-agents.

A. OMP

1) Order Management Agent: Is one that provides an interface to customers. In a foundry fab, the order management agent has to process the incoming orders, estimate the due date, provide the real-time information of orders and notify the exception of lot processing. In short, the order management agent should provide all the information of orders to customers. Since customers of the foundry fab are all of enterprise-scale, the order management agent should provide a standard interface for customers. It can be viewed as a Business-to-Business communication. As a result, eXtended Markup Language (XML) and Partner Interface Process (PIP) in the RosettaNet [19] are used in order management agent as the standards to communicate with customers.

2) Available-to-Promise Agent: The objective of the available-to-promise agent is to provide the confidence level for sales to receive orders. It takes the estimated cycle time of the orders and the available capacities of fab into account. Hence, sales can promise to on-time delivery of orders based on its result. The ways to estimate cycle time is based on the on-line learning agent.

B. Planning Process

1) Production Planning Agent: After receiving the orders, the production planning system should transfer the customer orders into manufacturing orders, i.e., the release schedule of lots. Time horizon in the production planning is a key attribute

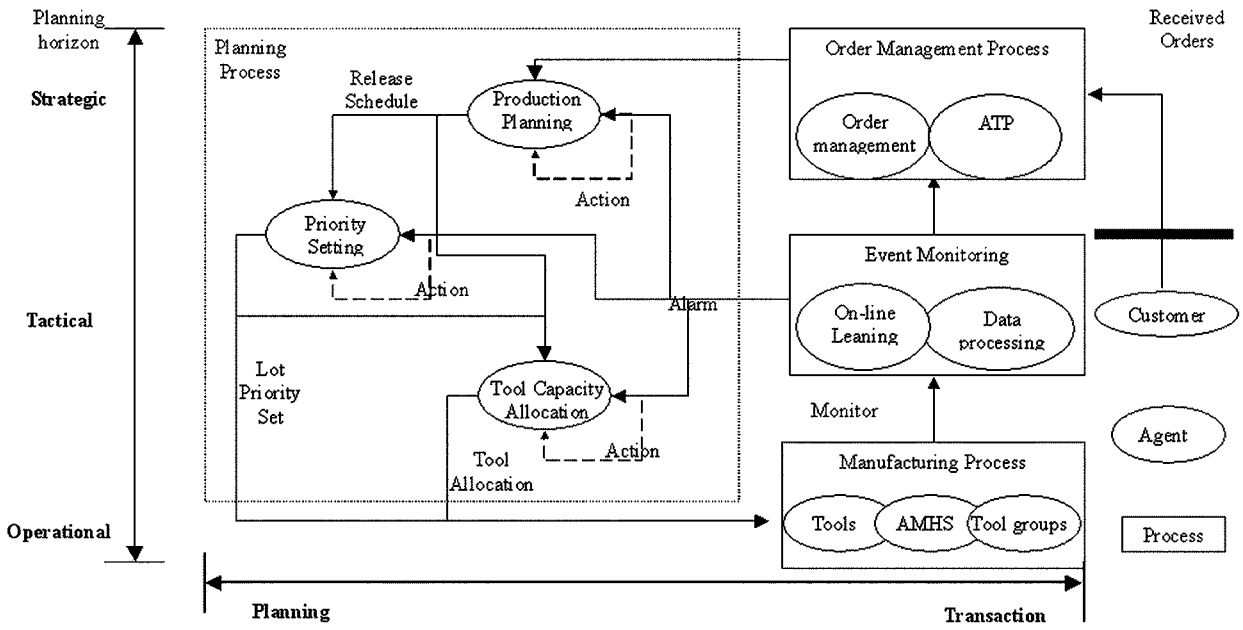


Fig. 5. The subprocesses and agents in the OFP.

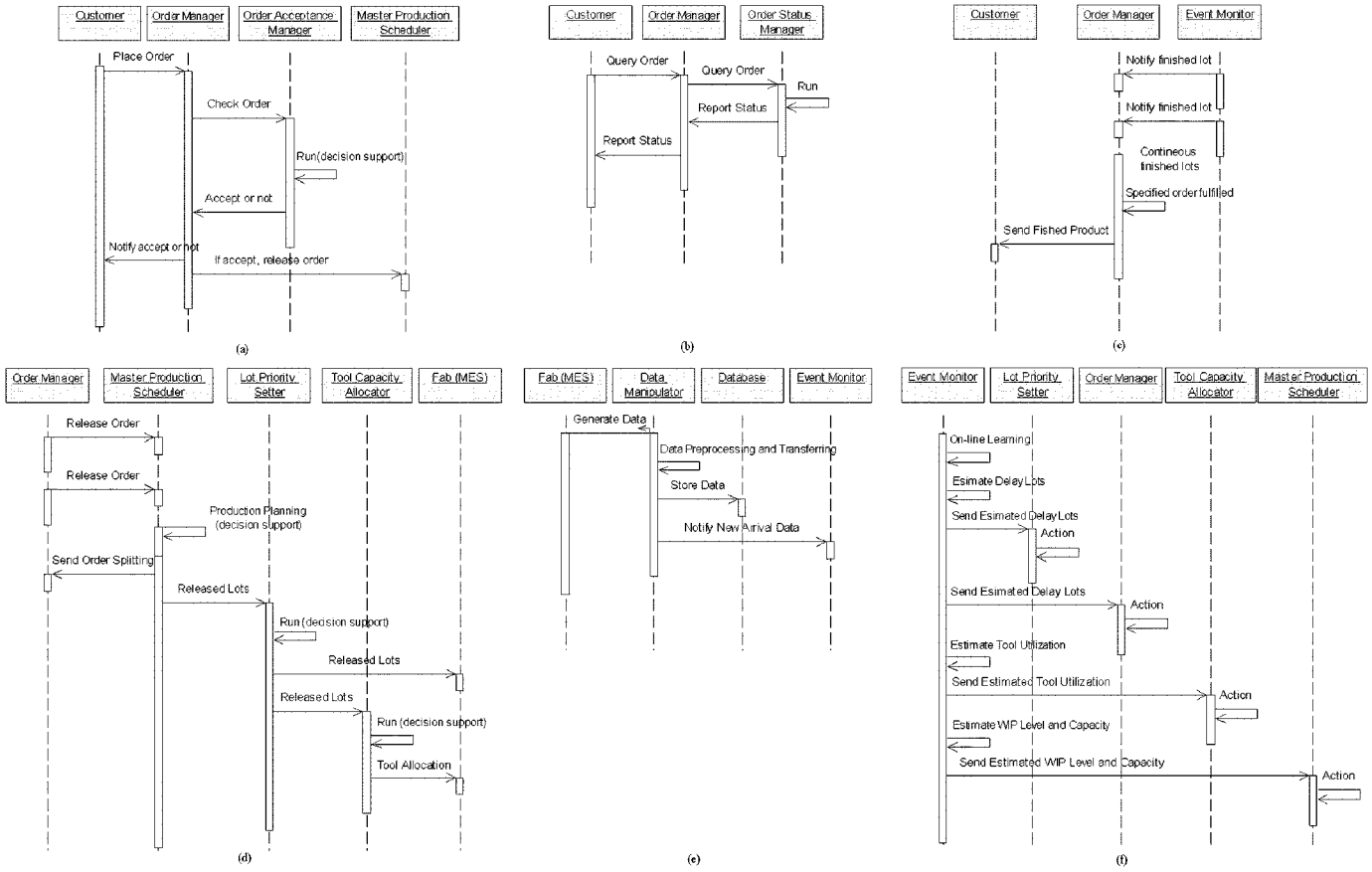


Fig. 6. Sequence diagrams for agents. (a) Order arriving. (b) Order querying. (c) Order finished. (d) Planning. (e) Data generating. (f) Event monitoring.

to decide the release schedule. If the time horizon is too short, the long-term order effects will not be considered. On the other hand, if the time horizon is too long, the planning system may be too complicated. Many lot release rules are summarized in [8]. Idle avoidance, Constant work-in-process (CONWIP), POISS, DETERMIN, Workload Regulating (WR) and PWR (Parametric WR) are frequently used as heuristic rules. WR is used as the lot release rule in this paper.

2) Priority Setting Agent: The released lot is assigned a priority class by the system. In order to have a better performance for on-time delivery, the least slack policy is used to prioritize all lots daily in the fab before executing the tool capacity allocation procedure. The proportion of each priority class is fixed as 5% lots for Super Hot lot, 5% for Hot lot, 30% for Rush lot, 45% for Normal lot, and 5% for Slow lot. The determination of the proportion is based on the research in [2] and the engineers' experiences from the real fab.

The estimated remaining cycle time and the slack value of each lot can be calculated from equations given in Section IV-D. If the slack $s(i) > 0$, it means that the lot commits its due date on time; otherwise, the lot will be delayed. The smaller the slack value of the lot has, the higher the priority class of the lot will be re-assigned. Since the proportion of each priority class is fixed, only partial lots have raised their priority classes. Therefore, the slack value of a lot is served as the reference for adjusting its priorities. The adjustment also takes the importance of lots into account. As a result, the slack value, current priority class and lot importance are the main features to decide the next priority class of a lot.

The steps to re-assign the priority class of each lot are listed below.

Step 1) Calculate the slack values of all lots in the fab;
Step 2) Highlight some important lots by the managers;
Step 3) Sort the lots in terms of the larger slack value and importance;
Step 4) Re-assign the priority class according to the proportion, $(SH, H, R, N, S) = (5\%, 15\%, 30\%, 45\%, 5\%)$.

3) Capacity Allocation Agent: The maximum number of wafers that a tool (i.e., semiconductor equipment) can process in a day is called the capacity of a tool. In general, the capacity of a tool depends on the preventive maintenance (PM) schedule, the frequency of the setup change, recipes and the idle time. It is difficult to estimate the actual capacity of the tool due to the unpredicted interrupt and the idle time of the tool.

The method for estimating the capacities of the tools in this paper is the dynamic moving average of the past seven days. Let $\mathrm{AC}_{T(n)}(t)$ be the actual capacity of the tool n at day t and $e_{T(n)}(t)$ be the utilization of the tool n at day t. The upper bound of the capacity of the tool n at day t, $\mathrm{UC}_{T(n)}(t)$, can be obtained by

$$\mathrm{UC}_{T(n)}(t) = \frac{\mathrm{AC}_{T(n)}(t)}{e_{T(n)}(t)} \qquad \forall T(n) \in T. \tag{1}$$

Then the estimated capacity, $\mathrm{EC}_{T(n)}(t)$, can be obtained by

$$\mathrm{EC}_{T(n)}(t) = \frac{\sum_{i=1}^{7} \mathrm{UC}_{T(n)}(t-i)}{7} \cdot \Omega_{T(n)} \qquad \forall T(n) \in T$$
$$\Omega_{T(n)} = \frac{24 \text{ hours} - \text{PM time period}}{24 \text{ hours}} \tag{2}$$

where $\Omega_{T(n)}$ is the portion of available time for processing.

The purpose of this agent is to decide the amount of the tool capacities for each tool group. For example, the upper bound capacity of the tool n is 300 wafers and the tool n is enlisted in two tool groups. After running the capacity allocation module, the system will reserve 100-wafer processing capacity for the tool group 1, 150-wafer capacity for the tool group 2 and 50-wafer capacity is not used.

The capacity allocation algorithm is given below.

Step 1: Load all necessary data
The WIP information, the release schedule, the capacities of the tools, the mapping table of the tools and the tool groups are loaded into the system.
Step 2: Sort the lots by their priority class
Apply the method mentioned in Section IV-B-II
For $i = 1$ to L (L: the number of all lots in the fab plus new released lots)
Step 3: Calculate the steps that lot i will go through in the day
Applying (4), which is given in Section IV-D
For $k = 1$ to $S(i)$
Step 4: Find the next tool group (NTG)
Looking up the route of lot i. Let NTG = TG(m)
Step 5: Find the candidate tool in the tool group
Applying Maximum Fuzzy candidate rule (MFCR) [18] to find the canditate tool.
Step 6: Allocate the capacities for the lots
If the remaining capacity of T^*, $\mathrm{RC}_{\mathrm{T}^*}$ > the number of the lot i, $N_L(i)$, then reserve the capacity for the lot i and $RC_{T^*} = RC_{T^*} - N_L(i)$
else the lot is blocked in this tool group due to lack of the capacity. Stop allocating the following steps.
End for-loop
End for-loop
End of the procedure

The results of this procedure are the WIP information for one-day later, reserved capacity of the tool for each tool group, the bottleneck tools and the bottleneck tool groups. The bottleneck tools and bottleneck tool groups can be easily found from the capacity allocation algorithm. Let $N_{\mathrm{B}}(m) =$ the number of blocked lots in the tool group m. It tells that $N_{\mathrm{B}}(m)$ lots are blocked and cannot continue their routes due to the lack of capacity of the tool group m. If $N_{\mathrm{B}}(m)$ is too large, this tool group is called a bottleneck tool group.

A bottleneck tool group results in the blockage of all the tools enlisted in this tool group. The bottleneck responsibility (BR) of the tool n is defined in [18]. Similarly, if $\mathrm{BR}(n)$ is too large, for example 10 lots, the tool n is called the bottleneck tool.

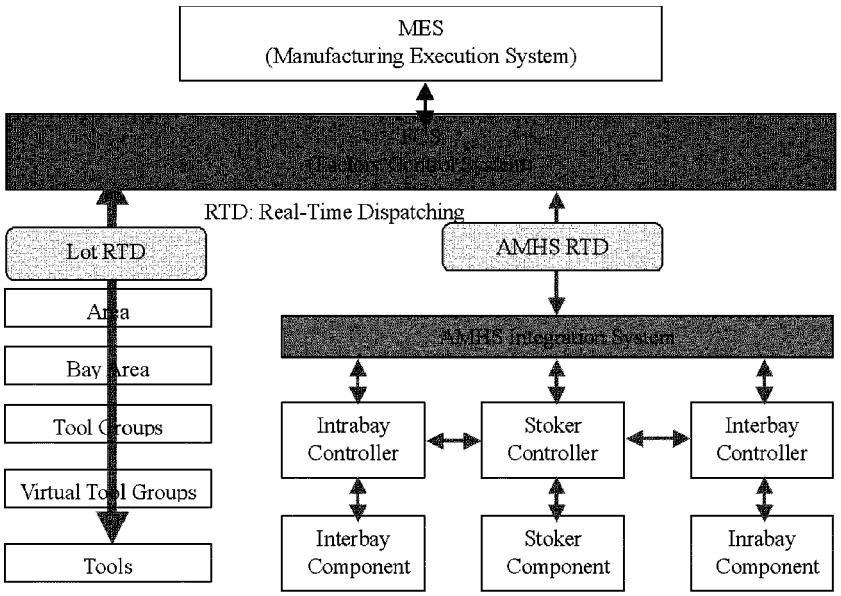


Fig. 7. The hierarchy of MES and shop floor elements.

The shop floor manager should take action to solve the bottleneck issue based on the information provided above. The action depends on the extra available resources, the experience of the managers, the PM schedule of the tools and the on-time delivery performance. Although it is not easy to make decision, the manager can adjust the capacities of the tools with the aid of the above results. Which tool should be added or which tool's PM schedule can be changed in order to solve the bottleneck can rely upon the list of the bottleneck tools and tool groups.

After adjusting the capacities and the usage of the tools, re-run the procedures until the balance between the tool utilization and bottleneck is achieved.

C. MES

The MES is the manufacturing unit in the enterprise. It contains material flow, lot processing and lot movement in the shop floor. Hence, modeling of the MES should include the scheduling of lots, lot dispatching, tool dispatching, material movement and coupling effect between tool groups and tools [18]. The overall MES is controlled by the factory control system (FCS), as shown in Fig. 7. Tool groups, virtual tool groups and tools are modeled to represent the processing steps while the Automated Material Handling System (AMHS), which includes intrabays, interbays and stockers, are used to move lots. To complete one step of operations, the lot should flow into the processing hierarchy and AMHS hierarchy one at each time.

D. Event Monitoring Process

1) Data-Processing Agent: Is one who converts the collected data from the MES into the physical meaning of the foundry fab for information query and preprocesses the data for the on-line learning agent. In the first part, the historical and real-time information of lots, orders, tools, and vehicles should be prepared for customers, engineers, managers, and decision support systems. Generally speaking, data-processing agent is the data source of decision support for other agents.

2) On-Line Learning Agent: Its purpose is to learn the processing time and waiting time of the lots in each step. However, the flows of the lots in an IC fab are like job shops but more complex. It is very difficult to identify the relationship between tools and lots; therefore, taking the entire fab as a unit for modeling is a tremendous task. The following shows the on-line learning agent briefly. The detailed information can be referred to [17].

- *Tool model*

  Tools are real working machines, while tool groups are man-defined or virtual units used to define the flow of a product. Usually, the tools that can do the same operation belong to a tool group. The flow of a lot or a product is the sequence of the tool groups that only defines the operation (or tool group) it takes. The flow of a lot is related to its product type or typically its route type. Although the lot is processed inside the tools, the route of the lot is the flow of the tool groups. As a result, the lot processed in the previous step only knows the next tool group to which it should go. The available tools in that tool group can serve this lot. If no one is available, this lot will wait in some place or we say that it will wait in a virtual tool group. Lots waiting in the same tool group will compete for the same resources, the tools.
- *The client/server architecture of the on-line learning system*

  The on-line learning system is used to implement the tool model based on back-propagation neural networks. Each tool model is on-line retrained periodically. The client/server architecture of the on-line learning system is shown in Fig. 8. The server program retrains the neural networks while the client program estimates cycle time, tool group moves, bottleneck, etc.
- *Inputs and outputs of tool models*

  To construct the waiting/processing time models, the first step is to determine the inputs of the network. Twelve attributes are chosen as BPNN inputs and the waiting time

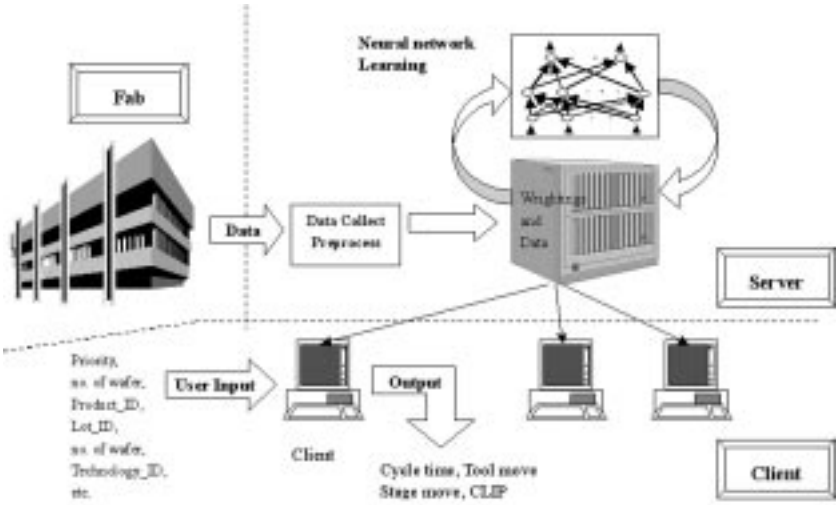


Fig. 8. The client/server architecture of on-line learning system.

as the output in the waiting time model. The inputs are waiting wafers, priority of the lot, push index, pull index, month index, day index, time index, and waiting pods of five priorities (SH, H, R, N, S). The output is the actual waiting time in minutes.

- *Support for other agents*

The client program uses the tool model to calculate the following information for other agents.

1) The cycle time of the lot i (CT(i)) is the summation of the time of each step. Namely,

$$\mathrm{CT}(i) = \sum_{n=1}^{N} (W_n + P_n) \tag{3}$$

where W_n is the waiting time of lot i in step n and P_n is the processing time of lot i in step n.

2) The number of steps, $S(i)$, that the lot will go through in a day can be obtained by

$$S(i) = \left\{ \arg_s \min \left\{ \sum_{n=k+1}^{s} (W_n + P_n) > 1 \text{ day} \right\} \right\} - k \tag{4}$$

where k is the current step and s is the step that the lot i will arrive one day later.

3) The estimated remaining cycle time (ERCT) of lot i based on least slack policy can be obtained by

$$\mathrm{ERCT}(i) = \sum_{i=m+1}^{N} (W_n + P_n) \tag{5}$$

where m is the current step and N is the total steps of the lot i.

4) The slack value of lot i, $s(i)$, is calculated as

$$s(i) = \delta(i) - t - \mathrm{ERCT}(i) \tag{6}$$

where $\delta(i)$ is the due date of lot i and t is the current time.

3) Event Monitoring Agent: The event monitoring agent notifies the other agents when the corresponding events occur. It will report the finished lots to order management agent, the delayed lots to priority setting agent, the bottleneck analysis to tool capacity allocation agent and the WIP level to the production planning agent.

V. Simulation and Result

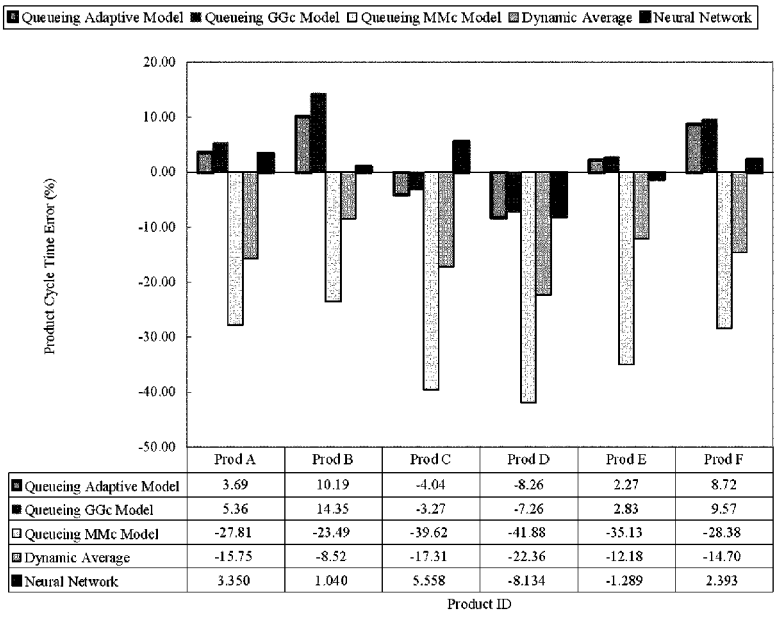
Each agent is constructed and performs its jobs on the GMPP. The simulation data are collected from a real foundry fab. There are six areas, about 250 tool groups, and 600 tools in the 200 mm fab. It takes three days to warm up the on-line learning agent. The simulation results include cycle time estimation, tool group move, tool move, and bottleneck identification.

A. Cycle Time Estimation

Three forecasting approaches are compared in the cycle time estimation, i.e., dynamic moving average, queueing model, on line agent in this paper. The following are the comparison results of the product cycle time and lot remaining cycle time forecast generated by each approach.

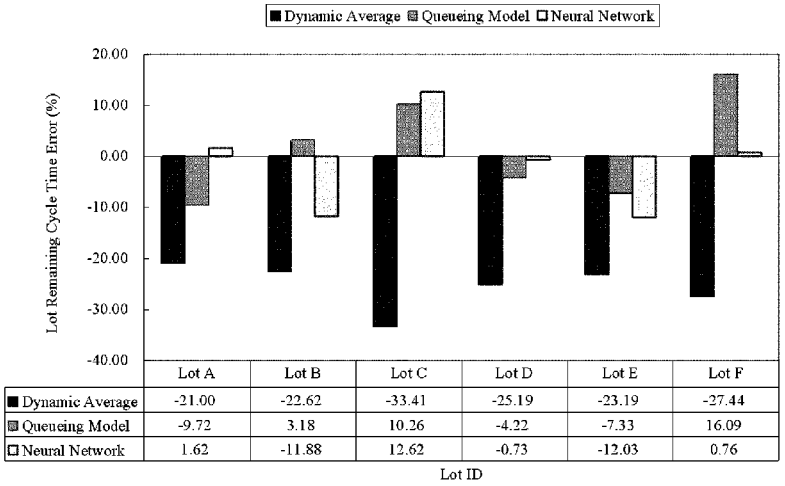
The product cycle time forecasting errors for each approach are shown in Fig. 9. Five lots are randomly selected to forecast the remaining cycle time of the lots, as shown in Fig. 10. In the queueing model, three different models are used. The most general and the most often used is MMc model. If the arrival and service patterns are approximately close to exponential distribution, MMc model will be a fast and good approximation. If the arrival and service pattern cannot be fitted to any kind of patterns or there is no specific formula suitable for all kind of circumstances, G/G/c is adopted. From the result, GGc model is more accurate than MMc model. The reason is due to none exponential distribution.

The error percentage of dynamic average method is larger than queueing adaptive model and neural network simply because the operation is modeled more roughly than the others. Since real-time information is updated from the real fab, the on-line learning by neural network has better modeling capability and longer computation than dynamic average method. Clearly, the on-line learning agent has better cycle time estimation than other approaches.



	Prod A	Prod B	Prod C	Prod D	Prod E	Prod F
Queueing Adaptive Model	3.69	10.19	-4.04	-8.26	2.27	8.72
Queueing GGc Model	5.36	14.35	-3.27	-7.26	2.83	9.57
Queueing MMc Model	-27.81	-23.49	-39.62	-41.88	-35.13	-28.38
Dynamic Average	-15.75	-8.52	-17.31	-22.36	-12.18	-14.70
Neural Network	3.350	1.040	5.558	-8.134	-1.289	2.393

Fig. 9. The comparison of product cycle times.



	Lot A	Lot B	Lot C	Lot D	Lot E	Lot F
Dynamic Average	-21.00	-22.62	-33.41	-25.19	-23.19	-27.44
Queueing Model	-9.72	3.18	10.26	-4.22	-7.33	16.09
Neural Network	1.62	-11.88	12.62	-0.73	-12.03	0.76

Fig. 10. The comparison of lot remaining cycle times.

B. Tool Group Moves

Tool group move, $\text{Move}_{\text{TG}}(m)$, is the throughput of the tool group in a day. When a lot completes an operation, the number of wafers of the lot is regarded as the moves in this tool group. For example, if a lot with 24 wafers has been processed by the tool group m, then 24 moves are added to the tool group m. Thus, we have

$$\text{Move}_{\text{TG}}(m) = \sum_{i \in \tilde{L}(m)}^{N} N_L(i) \tag{7}$$

where N is the number of tools in $\tilde{L}(m)$; $N_L(i)$ is the quantity of $L(i)$; $\tilde{L}(m)$ is the set of lots which are processed in the TG(m); and L is the set of all lots. Note that $\tilde{L}(m)$ is a subset of L.

The estimated total number of tool group moves in "1998-12-31" are 54 597 wafers, as shown in Fig. 11. The move of the tool group 33 is as high as 2850 wafer moves.

C. Tool Moves

Similar to the tool group moves, the estimated tool moves in the tool n, $\text{Move}_T(n)$, is the number of wafers that are processed in the tool n. We have

$$\text{Move}_T(n) = \sum_{i \in \tilde{L}(n)}^{N} N_L(i) \tag{8}$$

where $\tilde{L}(n)$ is the set of lots which are processed in $T(n)$.

Fig. 12 shows not only the allocated capacity but also the utilization of the tools. The utilization of the tool m is defined as

$$e_{T(n)}(t) = \frac{\text{AC}_{T(n)}(t)}{\text{UC}_{T(n)}(t)} \tag{9}$$

where $\text{AC}_{T(n)}(t)$ is the allocated capacity of the tool n at day t.

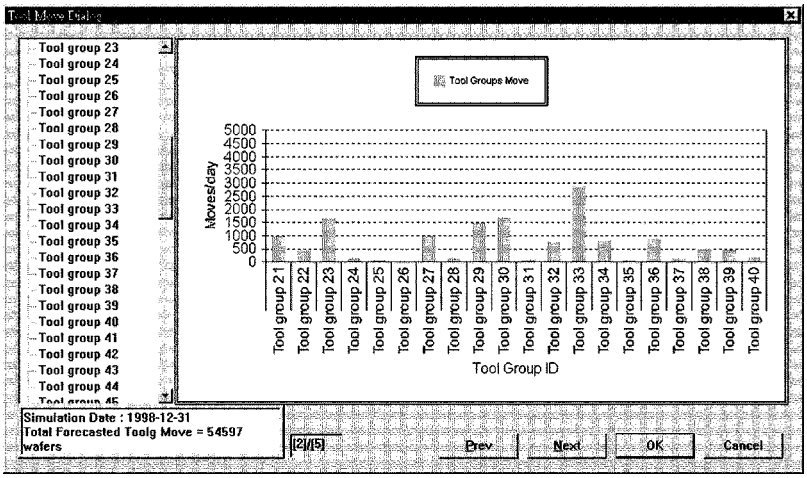


Fig. 11. Estimated tool group moves.

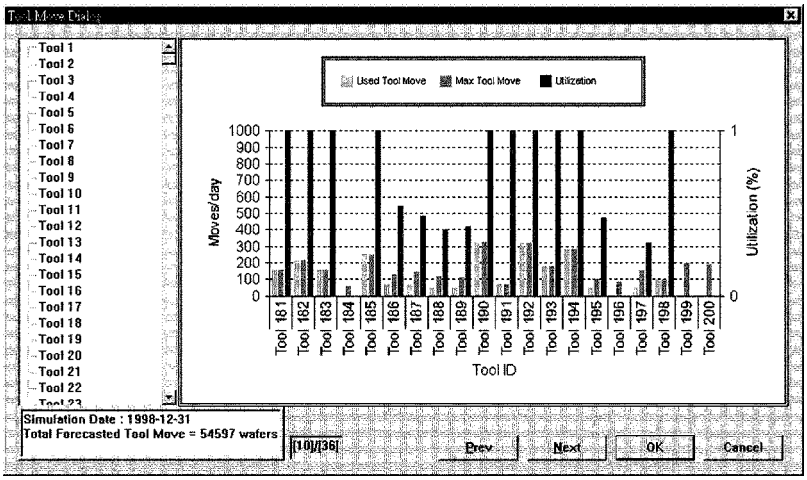


Fig. 12. Estimated tool moves.

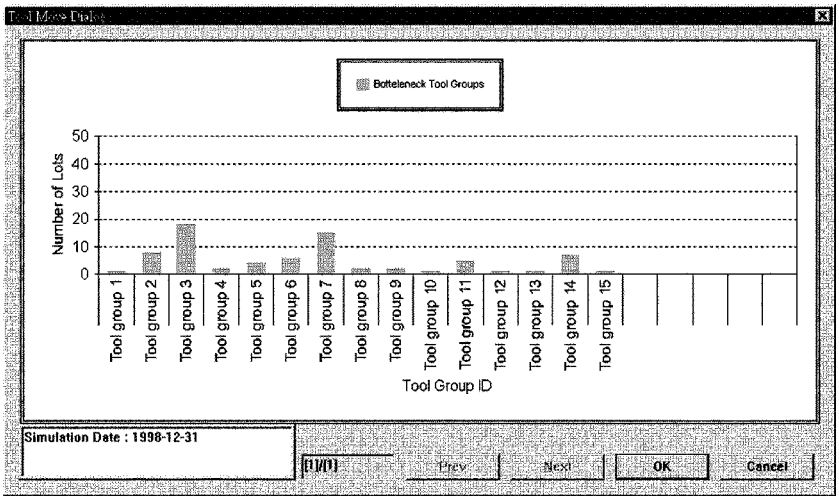


Fig. 13. Bottleneck tool groups.

D. Bottleneck Analysis

Bottleneck analysis is derived from the capacity allocation agent. Only the tool groups, which block the lots, are listed in Fig. 13. It can be found that the tool groups 3 and 7 are critical to the system. Over 10 lots are blocked in these two tool groups, which become the bottleneck tool groups.

In order to resolve the bottleneck tool groups 3 and 7, three and two new tools, for example, are added to the tool groups 3 and 7, respectively. The result shows that the bottleneck tool group 3 is no longer the bottleneck tool group, while the number of the blocked lots of the tool group 7 is reduced from 15 lots to 6. Note that it is only a reference for the manager. In general,

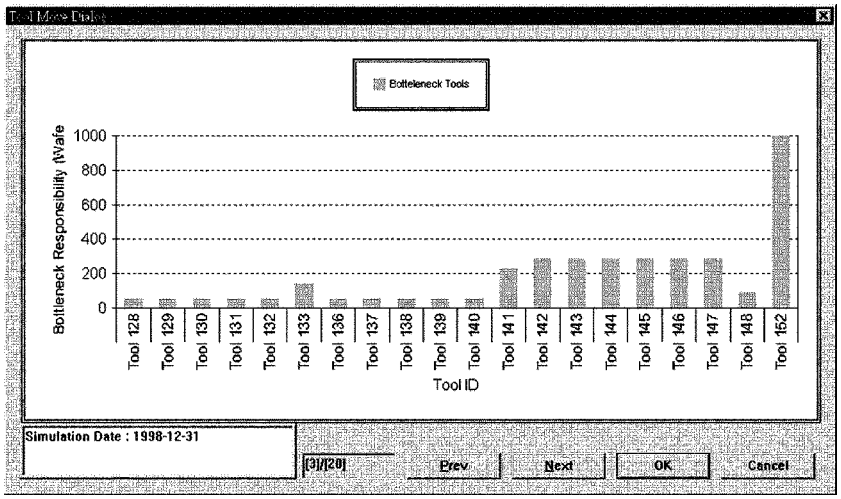


Fig. 14. Bottleneck responsibilities of tools.

the bottleneck tool groups can be handled by adding new tools or adjusting the PM schedules of the existing tools according to the actual data.

The bottleneck responsibility of tools is shown in Fig. 14. The responsibility of the tool 152 is 1104 wafers. The manager can identify the high-utilization tools and decide whether the PM schedule of the tool can be adjusted or not.

VI. Conclusions

The modeling of the OFP for the foundry fab is developed in this paper. The total OFP includes four sub-processes: order management, planning, manufacturing and event monitoring processes. Applications in the each process are constructed as agents in the distributed environment. Hence, the OFP is the enterprise-scale application integration. The contribution of this paper is that the supervisors can evaluate the rules in the OFP to detect the bottleneck of the tool groups and the delayed lots based on data provided by each agent.

A GMPP is proposed as the communication method within agents. The communication protocol on the GMPP can satisfy the requirements of ATA, ATP and PTA in a real-time, transparent, scalable distributed environment. In addition, the ESS is defined to provide the set of solution for users to choose.

Acknowledgment

The authors greatly appreciate the assistance from TSMC (Taiwan Semiconductor Manufacturing Company).

References

[1] W. Brenner, R. Zarnekow, and H. Witting, *Intelligent Software Agents: Foundations and Applications*. Berlin, Germany: Springer-Verlag, 1998.

[2] C. C. Cheng, "A Study of Lot Priority Setting for Wafer Fabrication," M.S. thesis, Inst. Industrial Eng., Chung Yuan Christian Univ., Chung-Li, Taiwan, R.O.C., 1998.

[3] P. Gullander, M. Fabian, S. A. Andreasson, B. Lennartson, and A. Adlemo, "Generic resource models and a message-passing structure in an FMS controller," in *Proc. 1995 IEEE Int. Conf. Robotics and Automation*, vol. 2, 1995, pp. 1447–1454.

[4] K. S. Ho and H. V. Leong, "An extended CORBA event service with support for load balancing and fault-tolerance," in *Proc. Int. Symp. Distributed Objects and Applications*, 2000, pp. 49–58.

[5] R. Kolluru, S. Smith, P. Meredith, R. Loganantharaj, T. Chambers, G. Seetharaman, and T. D'Souza, "A framework for the development of Agile Manufacturing Enterprise," in *Proc. 2000 IEEE Int. Conf. Robotics and Automation*, San Francisco, CA, 2000, pp. 1132–1137.

[6] C. H. Kuo, "Development of Distributed Component Based Manufacturing System Framework," Ph.D. dissertation, Dept. of Mech. Eng., National Taiwan Univ., Taiwan, R.O.C., 1999.

[7] L. R. Lin and H. P. Huang, "Real-Time networking for the implementation of CIM," in *Proc. Int. Conf. Automation Technology*, vol. 1, Taiwan, R.O.C., 1996, pp. 21–28.

[8] M. H. Lin and L. C. Fu, "Modeling, simulation and performance evaluation of an IC wafer fabrication system: A generalized stochastic colored timed petri net approach," *Int. J. Prod. Res.*, vol. 38, no. 14, pp. 3305–3341, Sept. 2000.

[9] F. R. Lin, G. W. Tan, and M. J. Shaw, "Modeling supply-chain networks by a multi-agent system," in *Proc. 1998 IEEE 31st Annu. Hawaii Int. Conf. System Science*, 1998, pp. 105–114.

[10] D. S. Platt, *Understanding COM+*, WA: Redmond, 1999.

[11] J. Siegel, *CORBA Fundamentals and Programming*. New York: Wiley, 1996.

[12] R. Silora and M. J. Shaw, "Coordination mechanisms for multi-agent manufacturing systems: Applications to integrated manufacturing scheduling," *IEEE Trans. Eng. Manage.*, vol. 44, no. 2, pp. 175–187, 1997.

[13] D. C. Sschmidt and S. Vinoski, "Overcoming drawbacks and decoupled communication in corba," in *C++ Rep.*, 1996, vol. 8, p. 10.

[14] Y. H. Su, R. S. Guo, and S. C. Chang, "Virtual fab: Enabling framework and dynamic manufacturing service provisioning mechanism," IEEE Trans. Eng. Manage., Jan. 2001, submitted for publication.

[15] G. S. H. Tan and K. L. Hui, "Applying intelligent agent technology as the platform for simulation," in *Proc. 31st Annu. Simulation Symp. 1998*, 1998, pp. 180–187.

[16] Y. Tang, M. C. Zhou, and R. Qiu, "Design of virtual production lines in back-end semiconductor manufacturing systems," in *Proc. 2000 IEEE Int. Conf. Systems, Man and Cybernetics*, Nashville, TN, Oct. 8–11, 2000, pp. 1733–1738.

[17] C. Y. Yu and H. P. Huang, "Fab model based on distributed neural network," in *Proc. Nat. Conf. Automation Technology*, ChiaYi, R.O.C., 1999, pp. 271–277.

[18] ——, "Priority-based tool capacity allocation in the foundry fab," in *Proc. 2001 IEEE Int. Conf. Robotics and Automation*, Korea, May 2001, pp. 1839–1844.

[19] [Online]. Available: http://www.rosettanet.org

[20] [Online]. Available: http://www.gsmworld.com

Chih-Yuan Yu was born in Chang-Hua, Taiwan, R.O.C., in 1975. He received the B.Eng. degree from the National Taiwan University, Taiwan, R.O.C., in 1997 and is currently pursuing the Ph.D. degree in mechanical engineering at the same university

His research interests include factory automation, enterprise integration, scheduling and dispatching, supply chain management, intelligent control, and distributed multi-agents.

Mr. Yu received the "Nomination for Kayamori Best Paper Award" at the 2001 IEEE International Conference on Robotics and Automation.

Han-Pang Huang (S'83–M'86) received the M.S. and Ph.D. degrees in electrical engineering from the University of Michigan, Ann Arbor, in 1982 and 1986, respectively.

Since 1986, he has been with the National Taiwan University, Taiwan, R.O.C., where he is currently a Professor in the Department of Mechanical Engineering and the Graduate Institute of Industrial Engineering. He was the Vice Chairperson of the Mechanical Engineering Department from 1992 to 1993, and the Director of Manufacturing Automation Research Technology Center from 1996 to 1999. Currently, he is the Associate Dean of the College of Engineering at the same university. His research interests include machine intelligence, network-based manufacturing systems, intelligent robotic systems, prosthetic hand, and nonlinear systems.