行政院國家科學委員會專題研究計畫　成果報告

以分散式代理程式系統建置與模擬虛擬晶圓廠的完整訂單流程(2/2)

計畫類別：個別型計畫
計畫編號：NSC91-2212-E-002-054-
執行期間：91 年 08 月 01 日至 92 年 10 月 31 日
執行單位：國立臺灣大學機械工程學系暨研究所

計畫主持人：黃漢邦

計畫參與人員：游志源　陳慶順

報告類型：完整報告
報告附件：出席國際會議研究心得報告及發表論文
處理方式：本計畫可公開查詢

中　華　民　國 92 年 10 月 27 日

行政院國家科學委員會專題研究計畫年度報告

以分散式代理程式系統建置與模擬虛擬晶圓廠的完整訂單流程 (2/2)

(Implementation and Simulation of a Virtual Fab based on Distributed Multi-agent Architecture)

計畫編號：NSC-91-2212-E-002-054-

執行期限：91 年 8 月 1 日至 92 年 7 月 31 日

主持人：黃漢邦 台灣大學機械工程學系

Email: hanpang@ntu.edu.tw

研究人員：游志源 陳慶順 台灣大學機械工程學系

一、中文摘要

本計畫主要建置一個虛擬的晶圓廠，用以模擬自接訂單到完成訂單的企業流程。由於系統規模大，本計畫將以分散式環境作為模擬與測試的基礎架構。整個系統包括下列模組：企業對企業的資訊交換介面、訂單管理系統、生產規劃、機台產能規劃、晶圓優先權設定規劃、製造執行系統、事件通報系統與中控系統。每個子系統將建置成代理程式(agent)架構在分散式的環境之中, 各自執行任務與達成目標。

在第二年中，我們將建置訂單管理系統與規劃系統。訂單將由訂單管理系統全權管理，包括接收、查詢和交單通知等。生產規劃的目的是將接收到的訂單轉為每日生產時的工單。晶圓優先權設定規劃系統是設定每日晶圓加工的優先順序以其提高達交率，進而提升客戶的滿意程度。機台產能規劃系統是考量現場在製品的狀況，進而動態調整機台的使用與預防保養的時程。本論文將提出一個高效率的 Maximum Fuzzy Candidative Ratio 作為的機台派工法則，結果顯示 MFCR 比其他法則能更有效的減少瓶頸機台與提升生產量。事件通報系統的任務是處理來自製造執行系統所產生的資料，配合決策支援系統將事件通報給相對應的代理程式來處理各種狀況。

關鍵詞：虛擬晶圓廠、產能分配、BizTalk、RosettaNet

Abstract

The key success of the foundry fab is based on the order fulfillment and the customers' satisfactory. The main objective of order fulfillment is to deliver the products on time. However, many processes, including due date setting, planning and scheduling of the order and real-time shop floor operations, are involved in accomplishing this goal. In order to enhance the customers' satisfactory, the foundry company should not only promise on-time delivery but also provide the real-time information of orders. Virtual fab (VF) can realize these two important issues.

The coupling effect is discussed in the system. Hence, a new tool-dispatching rule, Maximum Fuzzy Candidative Ratio (MFCR), is proposed to solve the capacity allocation problem under the constraint of the coupling effect. The results show that MFCR has better performance in reducing bottleneck tools and enhance the throughput than other rules. Besides, a Business-to-Business environment is built by applying the RosettaNet standard and the Microsoft BizTalk Server for the information exchange between the foundry fab and customers over the Internet.

Keywords: Virtual Fab, Capacity allocation, Priority setting, BizTalk, RosettaNet

二、計畫緣由及目的

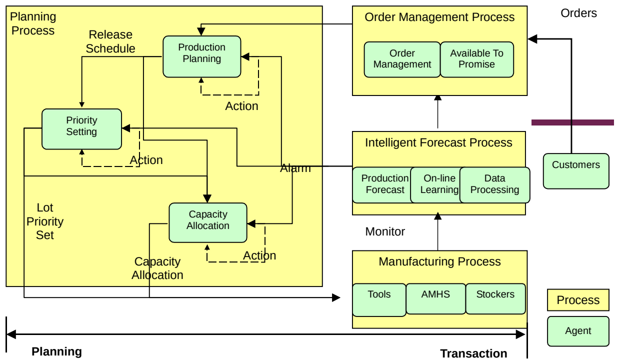

Fig. 1 The order fulfillment process

The core of virtual fab is the order fulfillment process. The process is triggered by the customers' orders in the foundry fab environment. This project will focus on the entire order fulfillment process. It includes order management process, planning processes, shop-floor operation, and event monitoring process, as shown in Fig. 1.

The major functions of the order management process are: 1) as customer portal, 2) due date quotation, 3) order status/tracking, and 4) order commitment.

The planning process includes production scheduling [1], capacity planning, lot priority setting and scheduling. The duty of production planning is to convert the orders into lot release schedule and to consider long-term capacity and WIP profile. The duty of priority management is to assign the priority value for all new released lot and re-assign the priority value for WIP. As to capacity planning, it should be responsible for tool allocation adjustment and re-allocation. The manufacturing process is modeled using Colored Timed Petri net to reflect the capacity constraints and dynamic behavior of lots. There are many researches [5-9] related to the modeling of the manufacturing execution system in the foundry fab.

The focuses of the first year are described below. The event monitor senses all real-time data to reflect alarm situation or trigger other processes to action. It can provide 1) lot priority decision support for priority management process, 2) tool configuration decision support for capacity planning process, 3) available-to-promise (ATP) decision support and order status decision support for order management process WIP level and capacity decision support for production planning.

The manufacturing process is triggered by the customers' orders in the foundry fab environment. The order is passed to the order management agent to decide the acceptance according to the capacity of the fab and Available To Promise (ATP) function [10,11]. If the order is received, it is passed to Production Scheduler (PS), which transfers the orders to the lot release schedule with the aid of the capacity planner. The fab model, including AMHS (automated material handling system), is constructed using Colored Timed Petri net (CTPN). Then the cycle time estimator and priority adjuster takes the real-time data to predict the cycle time of products for ATP function support and dynamically adjust lots' priorities for on-time delivery enhancement [12].

An interface will be defined to communicate with suppliers or customers to enable B2B integration and supply chain management, especially for information sharing and passing. The manager can simulate the virtual fab operation to check the bottleneck tools, capacity and on-time delivery performance. The customers can fetch the real-time information of their orders and monitor the status of lots to prevent the delay of orders through the web interface.

三、研究方法

The order management process is the process that involves the environment for business-to-business integration, the management of the orders, and the function of available-to-promise. RosettaNet standard and the BizTalk framework is set up for process integration between customers and the foundry fab. The order management process faces to customers. It plays an important role in satisfying the requirement of customers, such as order release, order query and order shipment.

Order Management Agent

The activities, such as order arrival, query order status, and order completion are handled in the order management agent. In fact, it is the window of the foundry fab to customers in a B2B environment. In tradition, the window of the company to customers is sales. The customers contact sales to place orders and the sales should check the available-to-promise of the orders to customers manually. With the B2B integration, an order management agent is required to handle the activities related orders automatically. The detailed descriptions of three activities are mentioned below.

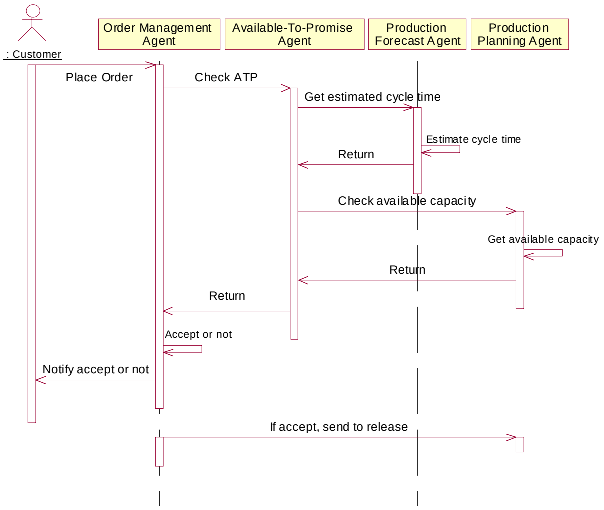

Fig. 2 Sequence diagram of the order arrival in the order management agent

Order arrival

When receiving a new order from a customer with Pip3A4PurchaseOrderRequest, the order arrival activity is triggered. The order management agent sends the order information to the available-to-promise agent to check the estimated due date and the availability of required capacities. The available-to-promise agent will get the estimated due date from production forecast agent and the available capacities based on the due date of the order from the production planning agent, respectively. If the estimated due date is larger than required due date or the capacities are not sufficient, the available-to-promise agent sends "Reject the order" back to the order management agent. Then the order management agent notifies the customer the denial of the order with Pip3A4PurchaseOrderConfirmation. On the contrary, the customer will receive the acknowledgement and the order is forwarded to the production planning agent for scheduling to produce. The sequence of the order arrival process is shown in Fig. 2.

Besides, the information of the estimated due date and the enough capacities for the order, the order management agent should also identify the customer after receiving the order, and deal with the cash and invoice after accepting the order. In this project, these two activities are not implemented without loss of the meaning of the order management agent. However, communication with other agents that are related to these two activities can be easily added to the GMPP platform.

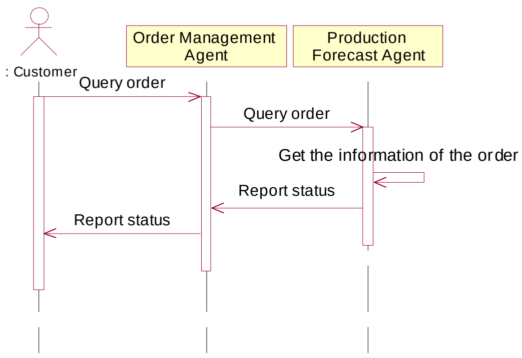

Fig. 3 Sequence diagram of the order query in the order management agent

Order query

In the supply chain, the statuses of orders are very important to customers. They can track the condition and the remaining time to complete their orders. Hence the order management agent should provide real-time information of orders to customers. As a result, the status of orders can be monitored in the customer's side over the Internet. The sequence of the order query activity is shown in Fig. 3. After receiving the request of real-time information about orders with 3A5PurchaseOrderStatusQuery, the order management agent asks the production forecast agent about the statuses of orders and then sends 3A5PurchaseOrderStatusResponse back to the customer. In the production forecast agent, it provides the information about the remaining cycle time of each lot, status of each lot and the current priority of each lot.

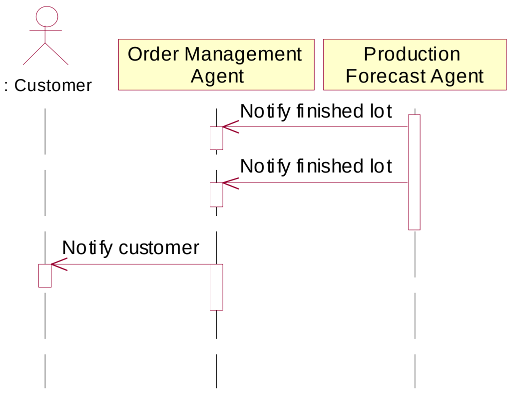

Fig. 4 Sequence diagram of the order completion in the order management agent

Order completion

When a lot completes its route, the production forecast agent will send a message to the order management agent. Until all lots of a certain order are completed, the order management agent should notify the customer for shipment by sending PIP3B2NotifyofAdvanceShipment. The sequence diagram of shipment notification is shown in Fig. 4.

Available-to-Promise Agent

The available-to-promise is the key agent in the customer-oriented company especially in the foundry fab owing to the characteristics of the make-to-order policy. Quick response to customers' orders and short cycle time are two key factors for a foundry fab to success. The responsibilities of the available-to-promise agent are to check the estimated due date and the available capacities for new orders. The sequence diagram order confirmation is shown in Fig. 4. After receiving the request of the order management agent, the available-to-promise agent sends the information of the order to the production forecast agent and retrieves the estimated due date of the order. The information needed to calculate the estimated due date includes the product type, quantity, the priority class that the customer assigns and the expected due date of the order. If the estimated due date is later than the expected one, the order should be rejected. Otherwise, the available-to-promise agent will ask the available capacities for the order from the production planning agent. The results of the expected due date and the availability of the capacities are sent back to the order management agent. They provide a basis for the judgment of the acceptance of the order.

On-time delivery plays the most important role in a foundry fab. In the strategy level of a foundry, while the acceptance of orders depend upon the company's collaborative relationship with the customers and the long-term capacity planning. In the tactical level, the major task of the manufacturing department is to fulfill all released orders. The supervisors always concern the contradictions between the WIP level and on-time delivery, and between the WIP level and the tool utilization. In general, the higher the WIP level is, the higher the holding cost of unfinished products and the worse performance of on-time delivery will be. On the contrary, if the WIP level is too low, the tool utilization may decrease due to lack of processing lots.

In order to achieve the on-time delivery performance, shorter cycle time, lower WIP level, higher tool utilization and throughput are required. Hence, the approaches to achieve the above goals are assigning the release policy, allocating tools' capacities, and deciding a best lot-dispatching rule. Hence, the production planning agent, the priority setting agent and the capacity allocation agent are created to fulfillment the above goals.

Priority Setting Agent

The functional view of the priority setting agent can be stated below. The inputs of the priority setting agent are the current status of lots. The outputs are the new priority set of lots. There are two important issues: (1) How many lots should be highlighted to re-set the priority? (2) How many degrees of a lot should be upgraded or downgraded? The former issue brings the consideration of the setting of d in CLIP(d) while the latter bursts out the choice of the setting rules. CLIP(d) means the estimated CLIP value that lots' due date is within the coming d days.

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

Step 1: Get the status of all lots

Step 2: Calculate the slack values of all lots

Step 3: Sort the lots in terms of the larger slack value and importance

Step 4: Re-assign the priority class according to setting rule.

Definition: The operator $p^N(i)$ devotes that the priority class of the lot *i* will be upgraded an integer *N* levels when *N* is larger than zero. When *N* is smaller than zero, the priority class of the lot *i* will be downgraded *N* levels. Note that the priority class must be laid between the lowest class and the highest class. The setting rules are described as follows.

Priority Setting Rule for Absolute Slack

The priority class of the delayed lot should be upgraded in order to speed up. However, when the lot is delayed too much, upgrading one level is not enough. Hence, a reference value to decide the level for upgrading and downgrading is created. The S(i) satisfies

$$p'(i) = \begin{cases} p^{-2}(i), & if \quad S_H \le S(i) \\ p^{-1}(i), & if \quad S_L \le S(i) < S_H \\ p^{0}(i), & if \quad -S_L \le S(i) < S_L \\ p^{1}(i), & if \quad -S_H \le S(i) < -S_L \\ p^{2}(i), & if \quad S(i) < -S_H \end{cases} \quad (1)$$

$$where \quad S_L > 0, \quad S_H > 0, \quad and \quad S_H > S_L$$

where S_L and S_H are given reference slack value to decide the priority class of the lot *i* and $S_L < S_H$. *p'(i)* is the adjusted priority class of the lot *i*.

Priority Setting Rule for Relative Slack

Sometimes, it is more urgent to apply the re-set priority on the lots whose due dates are approaching. Hence, the relative slack rule is derived from the absolute rule. *SR(i)* is defined as

$$SR(i) = \frac{S(i)}{ACT(i)} = \frac{ACT(i) - ERCT(i)}{ACT(i)} \quad (2)$$

Then the relative slack rue is defined as

$$p'(i) = \begin{cases} p^{-2}(i), & if \quad SR_H \le SR(i) \\ p^{-1}(i), & if \quad SR_L \le SR(i) < SR_H \\ p^{0}(i), & if \quad -SR_L \le SR(i) < SR_L \\ p^{1}(i), & if \quad -SR_H \le SR(i) < -SR_L \\ p^{2}(i), & if \quad SR(i) < -SR_H \end{cases} \quad (3)$$

$$where \quad SR_L > 0, \quad SR_H > 0, \quad and \quad SR_H > SR_L$$

where SR_L and SR_H are given slack ratio to decide the priority class of the lot i $SR_L < SR_H$.

Capacity Allocation Agent

Allocating the capacities of the tools

The main purpose of this section 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

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

$$Step(i,1) = \left\{ \arg\min_{s} \left\{ \sum_{i=k+1}^{s} (W_i + P_i) > 1\, day \right\} \right\} - k$$

For k = 1 to *Step(i,1)*

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 one tool-dispatching rule to find the candidate tool.

If the remaining capacity of T^*, RC_{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

Exit the following steps of allocating

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.

Maximum Fuzzy Candidative Ratio (MFCR)

MFCR is used in this project as the tool-dispatching rule. In the fuzzy system, there are two inputs and one output. Inputs are only chosen as $RC_{T(n)}$ and $DR_{T(n)}$ because $RC_{T(n)}$ and $RCR_{T(n)}$ are similar. The output is the fuzzy candidative ratio (FCR). MFCR means the tool with the maximum fuzzy candidative ratio is chosen to serve the lot in the tool group m and can be expressed by

$$T^* = \arg\max_T \{FCR_{T(n)}\} \qquad \forall T(n) \in T_{TG(m)},\ \forall RC_{T(n)} > 0$$

The fuzzy inference system contains the following properties.

- Linguistic variables {remaining capacity, dedicative ratio, candidative ratio}.
- Term set of remaining capacity T(RC) = { Too Small, Small, Large, Too Large }, Universe of Discourse U(RC) = [0, ∞] and membership function of RC is shown in Fig. 5.
- Term set of dedicative ratio T(DR) = { Too Low, Low, Medium, High }, Universe of Discourse U(DR) = [0,1] and membership function of RC is shown in Fig. 6.
- Term set of candidative ratio T(CR) = { Worst, Worse, Bad, Good, Better, Best }, Universe of Discourse U(CR) = [0,1] and membership function of RC is shown in Fig. 7.
- Fuzzy rules are shown in Table 1.
- The Mamdani fuzzy inference is adopted and uses product and max for T-norm and T-connorm operators, respectively. If RC=u_0 and DR=v_0, then the result CR' can be derived by

$$\mu_{CR'}(w) = \bigvee_{i=1}^{n} [\mu_{RC_i}(u_0) \cdot \mu_{DR_i}(v_0)] \cdot \mu_{CR_i}(w)$$

Centriod of area (COA) is used as the defuzzification strategy.

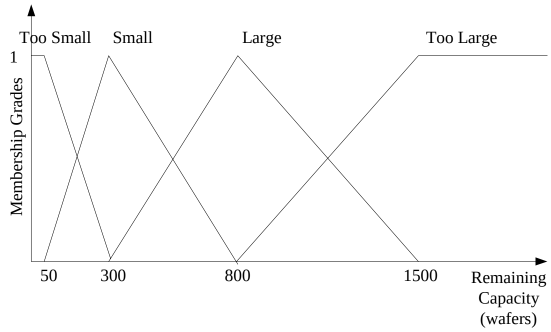

Fig. 5 Membership Function of Input 1

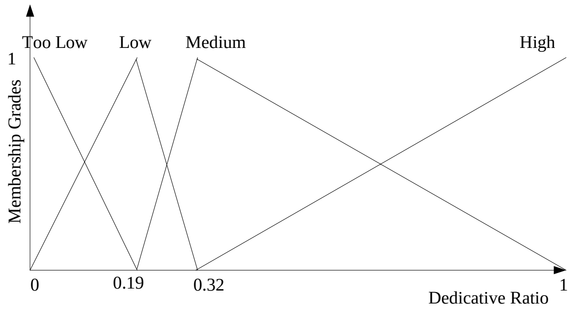

Fig. 6 Membership Function of Input 2

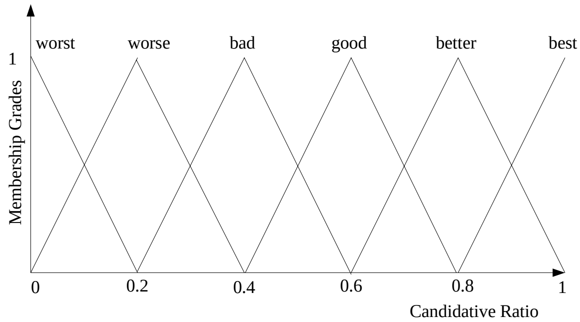

Fig. 7 Membership Function of Output

Table 1 Fuzzy Rules

		Remaining capacity			
		Too Small	Small	Large	Too Large
Dedicative ratio	Too Low	Worst	Worse	Worse	Good
	Low	Worst	Worse	Good	Good
	Medium	Worse	Bad	Good	Better
	High	Bad	Better	Better	Best

Fig. 8 Bottleneck tool groups

四、結論與成果

Bottleneck analysis

Only the tool groups, which block the lots, are listed in Fig. 8. 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 and they are called 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 more 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, 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.

Tool group moves

If the tool group move of the tool group m is $Move_{TG}(m)$, it means that there are $Move_{TG}(m)$ wafers flowed through the tool group m, i.e., the throughput of the tool group. The tool group move is defined as when a lot completes an operation, the lot contributes moves to this tool group. The moves that added to this tool group are equal to the number of the wafer of the lot. For example, if a lot with 24 wafers has been processed by the tool group m, then 24 tool group moves are added to the tool group m. Thus, we have

$$Move_{TG}(m) = \sum_{i \in \tilde{L}(m)}^{N} N_L(i) \quad (4)$$

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 $TG(m)$; L is the set of all lots. Note that $\tilde{L}(m)$ is the subset of L.

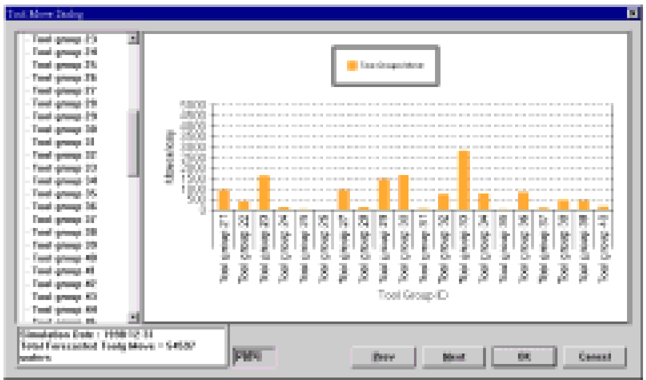

Fig. 9 Estimated tool group moves

The estimated total tool group moves in “1998-12-31” are 54597 wafers, as shown in Fig. 9. The move of the tool group 33 is as high as 2850 wafer moves.

五、參考文獻

[1] S. H. Chung, H. W. Huang, “The Design of Constraint-Oriented Target Planning System

for Wafer Fabrication Factories," Joint Conference of the 5th International Conference on Automation Technology, Taiwan, 1998.

[2] C.-H. Kuo, H.-P. Huang, "Failure Modeling and Process Monitoring for Flexible Manufacturing Systems Using Colored Timed Petri Nets," IEEE Transactions on Robotics and Automation, vol.16, no.3, pp.301-312, June, 2000.

[3] C.H. Lu, D. Ramaswamy, P.R. Kumar, "Efficient Scheduling Policies to Reduce Mean and Variance of Cycle-Time in Semiconductor Manufacturing Plants," *IEEE Transaction on Semiconductor Manufacturing*, vol.7, no.3, pp.374-388, 1994.

[4] M.A. Quaddus, "A Generalized Model of Optimal Due-Date Assignment by Linear Programming," *Journal of Operation Research Society*, vol.38, no.4, pp.353-359, 1987.

[5] A. A. Rodriguez and M. Kawski, "Modeling and Robust Control of Re-entrant Semiconductor fabrication Facilities: Design of Low-Level Decision policies," *A Proposal to Intel Research Council*, 1994.

[6] J. L. Snowdon and J. C. Ammons, "A Survey of Queueing Network Packages for the Analysis of Manufacturing Systems," *Manufacturing Review*, vol.1, no.1, pp.14-25, 1988.

[7] L. Sattler, "Using Queueing Curve Approximations in a Fab to Determine Productivity Improvements," *Proceedings of 1996 IEEE/SEMI Advanced Semiconductor Manufacturing Conference*, pp.140-145, 1996.

[8] K. S. Tsakalis, "Hierarchical Modeling and Control for Re-entrant Semiconductor Fabrication Lines: A Mini-Fab Benchmark," *IEEE/SEMI Advanced Semiconductor Manufacturing Conference*, pp.508-513, 1997.

[9] C.Y. Yu and H. P. Huang, "Fab Model Based on Distributed Neural Network," *National Conference on Automation Technology*, ChiaYi, Taiwan, pp.271-277, 1999.

[10] C.Y. Yu, H.P. Huang, J.Y. Juang, C.F Yeh, "Cycle Time Estimation for Highly Product-Mixed Semiconductor Foundry," *The First Cross Strait Workshop on Manufacturing*, Taipei, 2000.

[11] C.Y. Yu, S.W. Chen, H.P. Huang, "Estimation of Waiting Time for Products in an IC Fab, " *International. Conference on Automation Technology,* Taipei, Taiwan, 1998.

[12] C.Y. Yu, H.P. Huang, "On-Line Learning Delivery Decision Support System for Highly Product Mixed Semiconductor Foundry," *IEEE Transactions on Semiconductor Manufacturing*, vol. 15, no. 2, pp. 1-5, May 2002.