Realizing Dynamic Manufacturing Service Provisioning Mechanism in Order Commitment Service*

Shi-Chung Chang[1], Ruey-Shan Guo[2], Yea-Huey Su[2], Yi-Chang Lai[1]
[1]Department of Electrical Engineering, National Taiwan University, Taipei 106, Taiwan R.O.C.
[2]Graduate Institute of Business Administration, National Taiwan University, Taipei 106, Taiwan R.O.C.
scchang@cc.ee.ntu.edu.tw, rsguo@ccms.ntu.edu.tw, suesu@chopin.mba.ntu.edu.tw, yclai@ac.ee.ntu.edu.tw

Abstract *–In this paper, a dynamic manufacturing service provisioning mechanism (DMSPM) is designed to flexibly create and execute manufacturing services under a virtual fab (VF) enabling framework. The DMSPM exploits object-oriented (OO) technology to flexibly bind fab objects into services. It consists of four skeleton steps: name mapping, business process binding, resource reservation binding, and service management binding. To assess the feasibility and potential of DMSPM, the order commitment service (OCS) provided by a foundry serves as a study case. A prototype system is designed and implemented to realize DMSPM in the OCS application. The implementation demonstrated that current OO technology, CASE tools, fab information infrastructure and data/information availability make DMSPM readily realizable and that DMSPM has a good potential for application to virtual fab and e-business developments.*

Index Terms– *Virtual fab enabling framework, Dynamic manufacturing service provisioning mechanism (DMSPM), Order commitment service (OCS), Object-oriented (OO) technology*

INTRODUCTION

Fast growth of foundry business, fierce global competition and advancements of information technology have recently made the concept of "virtual fab" (VF) as one of the critical aspects for achieving competitiveness [1][2] in the semiconductor industry. One exiting definition defines VF as making a fab as if the customers' own fab [2]. It leads to a service-oriented new business model with emphasis on manufacturing service provisioning. Flexible and quick manufacturing service provisioning becomes a driving force for the development of a VF.

Manufacturing service provisioning includes the delivery of both tangible and intangible products. Tangible products are physical entities such as probed wafers or packaged dies, which represent a significant part of the value delivered to customers. Intangible products include serviced time, information and posession, which are values added to customers' needs. For example, after placing an order, a customer of a foundry fab would like to know specific production information before product delivery. To satisfy such a customer need, a foundry has not only to deliver the product on time but also to provide the customer with the necessary information.

To provide manufacturing services in a very dynamic business environment, the resources and business processes of a VF must be easily re-configurable in a systematic way. Exploiting the reusability, flexibility and scalability of object-oriented (OO) design methodology [9], Su et al [3] proposed a three-layer enabling framework for VF, where a fab is abstracted into three layers: manufacturing service (MS) layer, business process (BP) layer, and infrastructure (IR) layer. The three layers work together on a manufacturing service request but with different specialization. The MS layer deals with transformation of manufacturing services between the customers and the VF. The BP layer gathers all the objects needed to fulfill requirements from MS layer and controls both flows and rules of activities in order to facilitate manufacturing services provisioning. The IR layer consists of both physical and logical objects of a fab [5], which are cornerstones to support the upper two layers for manufacturing services provisioning.

Chang et al [4] have designed a dynamic binding mechanism for manufacturing service creation with guaranteed quality. It is designed on top of the OO model of a fab. A computer-aided software engineering (CASE) tool, Rational Rose, is adopted as an enabler of the dynamic binding mechanism, which supports a range of functions from OO model construction to C++ code generation. However, only have simple examples been previously studied to assess the feasibility of these ideas.

In this paper, by distincting the needs in service creation and execution phases, the dynamic binding mechanism is further refined into a dynamic manufacturing service provisioning mechanism (DMSPM). The order commitment service (OCS) provided by foundry companies such as TSMC and UMC is then extracted as a study case. Prototype implementation of is conducted to examine the feasibility, technological readiness and potetial of DMSPM for real application. Section 2 presents design of DMSPM under a VF enabling framework. An OCS example is described in Section 3. Protoype implementation is developed in Section 4. Finally, Section 5 concludes this paper.

DYNAMIC MANUFACTURING SERVICE PROVISIONING MECHSNISM

The dynamic manufacturing service provisioning mechanism (DMSPM) exploits the OO abstraction of a fab [5][6] and is service process driven. It is a control mechanism designed to enable the cooperation of objects among the three layers in the VF enabling framework of [3] to provide manufacturing services. Inputs to DMSPM are manufacturing service requirements from customers and its outputs are the managed manufacturing services. The key concept of DMSPM is object binding [7][10]. Upon a well-defined OO fab

* This work was supported in part by the National Science Council of Taiwan, Republic of China under grants NSC89-2212-E-002-040, NSC-90-2213-E-002-119, and NSC-89-2218-E-002-009

model, DMSPM utilizes binding in two phases. First, in the service creation phase, specific objects are *associated* with each other according to their static relationships to fulfil a requested service, where the static relationships among objects, either inter-layer objects or intra-layer objects, are built when modeling a fab. Second, in the service execution phase, those associated objects are *activated* by messages passing among objects to deliver the requested manufacturing service. After providing the service, the associated objects are released for other activations.

Manufacturing Service Provisioning Skeleton

A skeleton of the DMSPM is extracted from the procedural commonality of the two phases. There are four steps in the skeleton: name mapping, business process binding, resource reservation binding, and service management binding. Functions of the four steps are illustrated via the following descriptions about how the skeleton is applied to the two phases.

Steps of the creation phase:

- Name mapping: It automatically translates an external manufacturing service request into a specification expressed in internally recognizable terms and thereby creates an internal service requirement.
- Business processes binding: It associates business process objects according to the internal service requirement and thereby creates workflows, which are sequences of activities with various resource requirements.
- Resource reservation binding: It associates required resource objects to workflows. It reserves and schedules the use of necessary resources and thereby creates a fab service plan via functions such as resource registration, production planning and scheduling.
- Manufacturing service management binding: It associates managerial activities and resources to the fab service plan to it a managed service plan.

Steps of the execution phase:

- Business processes binding: It activates (binds) individual BPs and carries out the control of workflows according to the service plan.
- Resource reservation binding: It activates resource usage in accordance with the BP requirements.
- Manufacturing service management binding: It activates the managerial activities in each layer and thereby generates a managed manufacturing service to the customer.

Figure 1 illustrates how DMSPM is applied under a virtual fab enabling framework, i.e., how the skeleton steps just described bind objects of individual layers into a manufacturing service. In this Figure, an arrow represents the direction of interaction while the associated numbering represents the procedural sequence of DMSPM. For example, a customer requests manufacturing service via the interface object in the MS layer. After name mapping by such an interface object, DMSPM further binds BP objects in the BP layer accordingly to generate workflows for the manufacturing service. DMSPM then binds resources in IR layer to each BP object so on and so forth.

Note that DMSPM may provide flexible creation, execution and re-configuration of services under the VF framework in two ways. First, as an object can be easily modified, added into or deleted from a fab model, changes of a fab or customer requirements can then be captured into a service plan without changing DMSPM itself. Namely, DMSPM can easily respond to changes. Secondly, as one object is bound for activation only when needed, it allows binding by multiple services at different times and binding based on the actual progress of a service.

ORDER COMMITMENT SERVICE (OCS)

To convey the ideas and to assess the practicality of DMSPM, we adopt a simplified OCS as our case study. OCS aims at quick and accurate commitment of delivery dates once customers place their orders. Figure 3 depicts an abstract workflow of OCS, which is extracted from the practices of TSMC and UMC and is also locally called an available-to-promise (ATP) service. As shown in Figure 3, after receiving customers' orders, sales department sets priority to each order jointly with the production plan department (PP). According to this priority and ordered quantity of an order, PP makes the wafer output plan for the order by calculating capacity requirements of the order, checking residual capacity under the on-going production schedule, allocating the available capacity and scheduling resource reservation and the due date. The workflow finally confirms the due date with individual customers and then outputs results to complete OCS procedures. A good OCS must provide customers with quick and credible delivery schedules and transparent OCS business processes.

Figure 4 indicates a three-layered, object-oriented abstraction of OCS workflow in Figure 3. For examples, the sales-staff object and planning-tool object are abstracted from real entities of a fab. The OCS-BP object models the intangible OCS workflow and wafer-out-plan-BP object models the sub-flow of OCS workflow. Priority-business-rule and resource-reservation objects model logical business rules and algorithms, respectively. The static relationships among objects are also indicated in Figure 4, where a solid line indicate the existence of a relationship between two intra-layer objects while a dashed line for relationship between two inter-layer objects. For instance, an OCS-MS object in the MS layer involves both backend-outsourcing-BP object and OCS-BP object, which consist of other BP objects such as wafer-out plan-BP, confirmation-BP, and business-rules objects. Resource objects such as PP-staff objects in the infrastructure layer are needed to accomplish activities of wafer-out-plan-BP object in BP layer. Compared with functional description in Figure 3, the OCS model under the VF framework (Figure 4) is clearly superior in flexibility for service creation, execution and re-configuration.

PROTOTYPE OF DMSPM IN OCS

To realize DMSPM in the OCS application, a prototype system is designed and implemented. Figure 2 depicts the system architecture, where the DMSPM server is the emphasis of this implementation. The OCS model is expressed in terms of unified modeling language (UML) [8][9]. Rational ROSE is adopted as our CASE tool for both OO modeling and C++ code skeleton generation. The DMSPM server is coded in C++ while the browser and the webserver in Java.

The DMSPM server implements the common steps of DMSPM skeleton for creation and execution phases for OCS provisioning. Figure 5 indicates the detailed service processes. In the creation phase, a customer object requests OCS by sending messages of order specification to an user-interface object in the MS layer. The user-interface object then conducts name mapping and sends internal service requirements to OCS-MS object in MS layer. This OCS-MS object associates the necessary BP layer objects. As part of their functions, individual BP objects then bind necessary resource objects and reserves them in the infrastructure layer. An OCS plan is thus formed. Manager objects are then bound to the plan to check whether the service plan is feasible or needs revision. Although the manager objects are C++ programs in the prototype, they can be human decisionmakers in real application. The execution phase follow similar ideas and steps. Figure 6 depicts an object-centered view of DMSPM in OCS provisioning.

Among all the binding steps of DMSPM in OCS provisioning, the resource reservation step is the most computation intesive and an efficient decision engine is needed. In our prototype, without resorting to a commercial package, a resource reservation algorithm is designed and coded in Borland: lot allocation, capacity requirement calculation, residual capacity calculation, and capacity allocation. The "lot allocation" allocates lots belonging to firm orders with the same product specification. The "capacity allocation" function allocates the available capacity to individual lots. It evaluates the due date and wafer start date by using a PULL procedure to see if a lot can be delivered on time. If the due date of a lot cannot be met due to insufficient capacity, a PUSH procedure is then used to estimate a feasible delivery date by starting the lot as early as possible. There is no difficulty to replace this resource reservation module with a commercial package.

In our prototype development, it is found that input data and/or information to DMSPM are mostly available from the information system of a fab and its company. Figure 7 shows one kind of the inputs and outputs (including confirmation and schduling results) for OCS in a browser screen to customers. The inputs of OCS at least include order quantity, product type, technology, and expected due date while the output includes the committed due date and the wafer out plan. Some outputs of our experimentation are illustrated in Figure 8. Our experimentation results demonstrate both the ideas and the potential of DMSPM for application to virtual fab and e-business developments.

CONCLUSIONS

In this paper, a dynamic manufacturing service provisioning mechanism (DMSPM) has been designed for the creation and execution of manufacturing services in a virtual fab (VF). DMSPM has been designed to flexibly bind fab objects into services based on the object-oriented methodology. The prototype implementation of DMSPM in the OCS application has demonstrated that the ideas are feasible under current fab practice of information technology and that DMSPM should have a good potential in application to virtual fab and e-business developments.

REFERENCES

[1] Semiconductor Industry Association (SIA), *The National Technology Roadmap For Semiconductor*, San Jose: SIA, 1994.

[2] F. C. Tseng, "TSMC's semiconductor strategy (invited)," Keynote speech of *Proceedings of the Seventh International Symposium on Semiconductor Manufacturing*, pp. 5-7, Tokyo Japan, October 7~9, 1998.

[3] Y. H. Su, R. S. Guo, S. C. Chang, T. L. Chou, Y. C. Lai, "Manufacturing Service Creation and Management for Next Generation Virtual Fab," *Proceedings of International Symposium on Semiconductor Manufacturing 1998*, Tokyo, Oct. 7-9, 1998, pp.27~30.

[4] S. C. Chang, T. L. Chou, R. S. Guo, Y. H. Su, L. L. Lu, "A Dynamic Binding Model for Service Creation in Virtual fab," *Proceedings of 1998 Semiconductor Manufacturing Technology Workshop*, Hsinchu Taiwan, June 16-17, 1998, pp. 131-137.

[5] J. McGehee, J. Hebley, and J. Mahaffey, "The MMST computer-integrated manufacturing system framework," *IEEE Transactions on Semiconductor Manufacturing*, vol.7, no.2, pp. 107-115, May 1994.

[6] ESPRIT/AMICE, *CIMOSA: Open System Architecture for CIM*, Berlin: Springer-Verlag, 1993.

[7] A. A. Lazar, K.S. Lim, and F. Marconcini, "Realizing a foundation for programmability of ATM networks with the binding architecture," *IEEE Journal on Selected Areas in Communications*, vol.14, no.7, pp. 1214-1227.

[8] Quatrani, *Visual Modeling with Rational Rose and UML*, Addison-Wesley Publishing Company, 1998.

[9] J. Rumbaugh, I. Jacobson, and G. Booch, *The Unified Modeling Language Reference Manual*, Addison-Wesley, 1999.

[10] Bruce Eckel, *Thinking in Java* (2nd ed.), President, MindView, Inc., 2000.

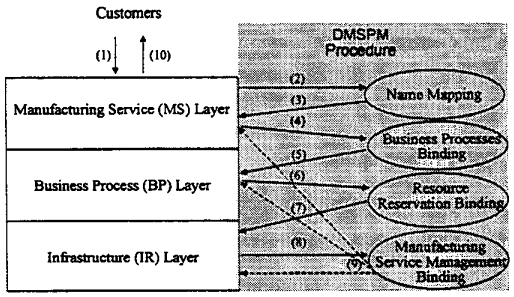
Customers
(1)
(10)
DMSPM Procedure
Manufacturing Service (MS) Layer
Business Process (BP) Layer
Infrastructure (IR) Layer
(2)
(3)
(4)
(5)
(6)
(7)
(8)
(9)
Name Mapping
Business Processes Binding
Resource Reservation Binding
Manufacturing Service Management Binding

Figure 1: Application of DMSPM on VF Framework

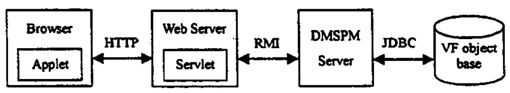

Figure 2 Prototype system of DMSPM in OCS

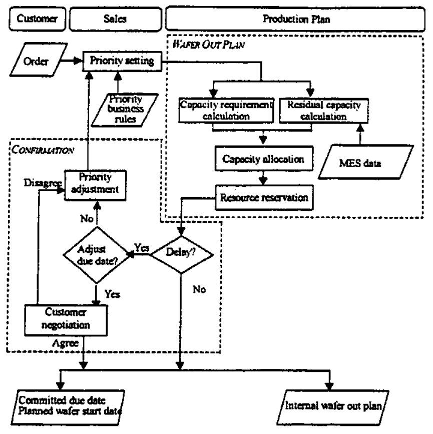

Figure 3 Order commitment service workflow

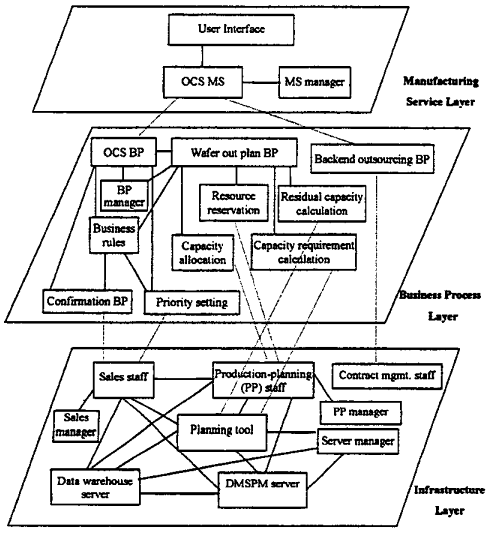

Figure 4 Relationships among objects for OCS provisioning

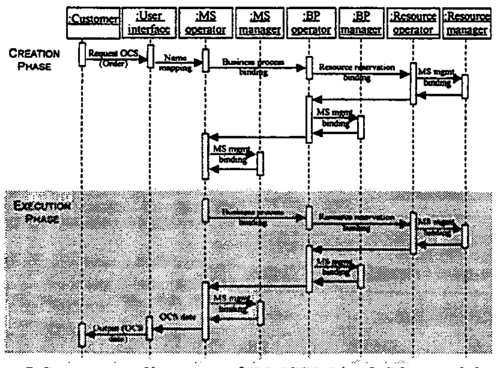

Figure 5 Sequence diagram of DMSPM in OCS provisioning

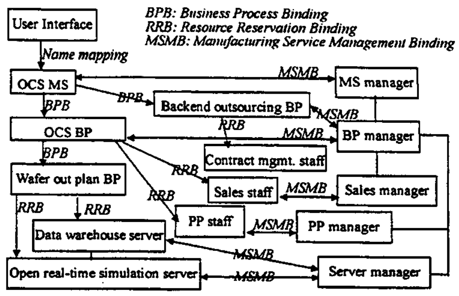

Figure 6 OCS architecture applying DMSPM

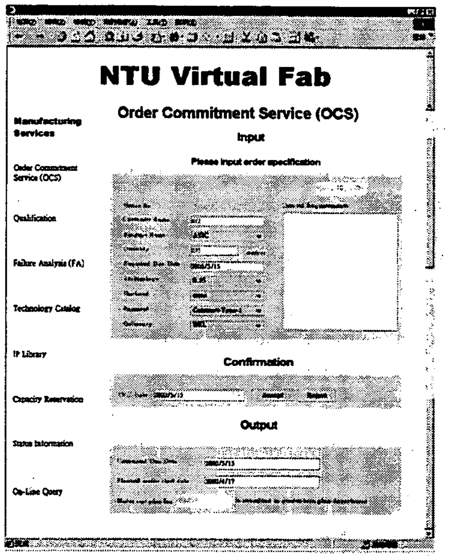

Figure 7 Input/output of OCS in browser

- Case I
  - Binding Service Creation
  -

Tech.	0.25 μm	0.28 μm	0.35 μm	0.25 μm	0.28 μm	0.35 μm
Request Due	20	20	20	10	10	10
Forecasted Due	20	20	20	13	10	10
W/S Date	8	10	13	1	1	3

- Case III
  - Bottleneck : Station 5, 6, 7

Lots	20	35	50	100	120
Capacity Allocation	PULL	PUSH	PUSH	PUSH	PUSH
Forecasted Due	15	16	18	22	24
W/S Date	2	1	1	1	1

Figure 8 Experimentation with Prototype System