國立交通大學

工業工程與管理學系

博士論文

半導體後段廠協同生產規劃之研究

Collaborative Production Planning for Semiconductor Backend Turnkey Service: An Activity-Based Costing Approach

研 究 生： 徐賢斌

指導教授: 蘇朝墩 教授

洪瑞雲 教授

中華民國九十三年十二月

半導體後段廠協同生產規劃之研究

Collaborative Production Planning for Semiconductor Backend Turnkey Service: An Activity-Based Costing Approach

研 究 生：	徐賢斌	Student: Hsien-Pin Hsu
指導教授：	蘇朝墩　博士	Advisor: Chao-Ton Su
	洪瑞雲　博士	Ruey-Yun Horng

國立交通大學

工業工程與管理系

博士論文

A Dissertation

Submitted to Department of Industrial Engineering and Management

College of Management

National Chiao Tung University

In partial Fulfillment of the Requirements

For the Degree of

Doctor of Philosophy

In Industrial Engineering and Management

December 2004

Hsinchu, Taiwan, Republic of China

摘　要

目前半導體產業已步入微利時代，後段產業環境更形嚴峻，個廠的單打獨鬥已難以生存。因此，半導體後段產業思考與他廠策略性聯盟，或經由整合內部各廠，以協同規劃 (Collaborative Planning)、協同運作 (Collaborative Opeation) 的手段，來提供一元化服務 (Turnkey Service)，並提升競爭力的作法，儼然蔚為風尚，成為永存續經營之思考方向。不過，有關協同規劃之研究文獻付之闕如，因此本研究致力於此項議題，提出一個融合作業基礎成本(Activity-Based Costing, ABC) 模式觀念的協同生產規劃系統 (Collaborative Production Planning System, CPPS)，本文中簡稱 ABC/CPPS，該系統在半導體後段廠協同生產環境下，可輔助生產規劃人員進行協同生產安排。

本論文首先進行協同規劃的整合性觀點以及 ABC 的理論探討，然後進行半導體後段廠(包含封裝及測試)的作業分析 (Activity Analysis)，經過作業的程序及轉變情形的進一步分析後，可獲得作業移轉圖 (Activity Transition Diagram)，接著將模式核心與作業移轉圖以述詞/擬轉網路 (Predicate/Transit Net, Pr/Tr Net) 來構建模擬模式，最後再將述詞/擬轉網路轉為邏輯程式語言並以 Visual Prolog 實作，則可完成 ABC/CPPS 構建。ABC/CPPS 主要由使用者介面 (User Interface, UI)、知識庫 (Knowledge Base, KB) 及模擬模式 (Simulation Model) 三個模組所構成，經由模擬後，協同生產排程計劃可規劃出來，以作為合作伙伴們共同協同執行之依據。簡言之，該模擬模式是整合了作業基礎成本 (ABC)，以及考慮了現場的資源動態限制而完成。本系統經由作業成本的導入模擬估算後，一個相對的利潤指標及協同生產排程可以提示出來，有了這個財務性指標，在進行生產規劃時，虛擬企業得以評估預期獲利情形。經由利潤的表示，企業上下階層方有共同一致的語言，因而更能預視 (透視) 永續經營的契機。

關鍵詞: 作業基礎成本，協同生產規劃系統，半導體後段廠，一元化服務，述詞/擬轉網路

ABSTRACT

While the market turns to an environment with low profit margins for semiconductor backend operations, it is hard for an independent firm to survive today. Forming strategic alliances or integrating an enterprise's internal firms by means of collaborative planning/operations to gain competitive advantage is inevitable. This study presents the development of an Activity-Based Costing Collaborative Production Planning System (ABC/CPPS) to help production planners to estimate the manufacturing profit of semiconductor backend turnkey (combined IC assembly and testing) operational service at the early stage of order release to production line in a collaborative context. The profit estimation is under the real constraints of production resources. First, the activities and their resources usage in semiconductor backend manufacturing were analyzed. Then, the flow of Manufacturing Orders (MOs) among activities was traced to obtain the "Activity Transition Diagram". Finally, a comprehensive Predicate/Transition Net (Pr/Tr Net) is used to simulate and implement the Activity-Based Costing (ABC) model with the dynamic characteristics of a production line incorporated. A financial measure, profit, is used to supplement and indicate the consequence of the planning result and link the view to the enterprise's financial vision. By implementing the ABC/CPPS, with Visual Prolog, into a rule-based simulation model, an expert planning system is built which composed of User Interface (UI), Knowledge Base (KB) and the simulation model (Model) core components, it seems well suited for collaborative production planning for semiconductor backend turnkey service. A numerical example is provides for detailed illustration.

Keywords: Activity-Based Costing (ABC), Collaborative Production Planning System (CPPS), Semiconductor backend, Operational Turnkey Service, Predicate/Transition Net

誌　謝

三年多來的博士學程，如今完成，本論文得以順利出爐，要感謝的人太多了，如果沒有他們的陪伴及鼓勵，孤寂的道路難見盡頭。在沿路途中，有靈光乍現的喜悅，有挫折及困境，所幸在師長、親友的鼓勵協助下，都逐一克服，一一渡過。所有過去的一切，如今全部化作為一顆感恩的心。本著雖非曠世絕作，但絕對是這些幕後曾經付出關懷、溫情者其用心及精神的最佳見證。

首先要感謝指導教授蘇朝墩博士三年多來的悉心指導，從本文的研究、想法、撰寫到發表，給與學生發展的空間、尊重及指教，使我獲益良多，謹此致上衷心的敬意和感謝。還有，感謝洪瑞雲博士對本人在困境中的及時協助及鼓舞，對其感謝及敬意長銘在心。再者，交通大學工業工程與管理系梁高榮博士、大業大學工業工程系駱景堯博士和台北科技大學經營管理系陳穆臻博士於論文審查期間不吝提供之寶貴的意見，使得論文內容得以更臻完善，在此致上最深的謝意。此外，感謝摯友台林、志華、俊欽及明新企管系老師們的鼓勵，使得學業得以順利完成。

最後，謹以此論文獻給我最親愛的親人；我的雙親及妻女，麗光及彩庭。感謝父母的培育及內人的付出，使我能夠無後顧之憂的專心研究。最後，真心祝福陪我渡過這段成長歲月，所有關心我與我關心的人。

CONTENTS

Page

摘要 I

ABSTRACT II

誌謝 III

CONTENTS IV

LIST OF FIGURES VI

LIST OF TABLES VII

CHAPTER 1 INTRODUCTION 1

1.1 Research Motivation 1

1.2 Research Objectives 4

1.3 Scope and Assumptions 4

1.4 Organization 6

CHAPTER 2 COLLABORATIVE PRODUCTION PLANNING FOR SEMICONDUCTOR BACKEND ENTERPRISE 7

2.1 Definition and Types of Collaborative Operation 7

2.2 The Key Issues for Collaborative Operation 9

2.3 A Collaborative Production Planning System Structure 12

CHAPTER 3 METHODOLOGY 14

3.1 Review of the Decision Models 14

3.2 The ABC/M Model 15

3.3 The ABC/CPPS 18

CHAPTER 4 DEVELOPMENT OF ABC/CPPS 26

4.1 Analyze the ABC Data 26

4.1.1 Define the Domain of the Application 26

4.1.2 Understand the Process of Semiconductor Backend Operation 26

4.1.3 Analyze the Activities/Resources Form the Process 31

4.1.4 Analyze the Cost Driver and Cost Object 32

4.2 Development of the ABC Pr/Tr Net 35

4.2.1 Analyze the Activity Transition 35

4.2.2 Group the Activities into Activity Sets 34

4.2.3 Develop the ABC Pr/Tr Net Simulation Model 37

4.2.4 Profit Calculation 39

CHAPTER 5 IMPLEMENTATION OF ABC/CPPS FOR SEMICONDUCTOR BACKEND TURNKEY OPERATION 41

5.1 Design the Data Structure 41

5.2 Implementation of the ABC Pr/Tr Net 43

5.3 System Structure of ABC/CPPS 51

5.4 Web Publish and Data Exchange 52
CHAPTER 6 NUMERICAL EXAMPLE 54
6.1 The Scenario 54
6.2 The Input Data 59
6.3 The Outcomes 59
CHAPTER 8 CONCLUSIONS 60
7.1 Conclusion 60
7.2 Future Work 62
APPENDIX A COLLABORATIVE SCHEDULE FOR ASSEMBLY FIRM ASE 64
APPENDIX B COLLABORATIVE SCHEDULE FOR TESTING FIRM ASET 67
APPENDIX C COLLABORATIVE SCHEDULE FOR TESTING FIRM WTAE 68
APPENDIX D IMPLEMENTATION OF XML DATA EXCHANGE 70
D.1 XML and Data Exchange Flow 70
D.2 DTD Introduction 71
D.3 XML Document 72
D.4 XML Data Conversion 74
D.4.1 Understanding the Data Structure 74
D.4.2 Design of the DTD Used in XML Document 75
D.4.3 Development of XML Document Generating Engine 78
REFERENCES 80

LIST OF FIGURES

Page

Figure 1.1 The performance pyramid ··· 3
Figure 1.2 Supply chain of semiconductor industry ··· 5
Figure 2.1 Conceptual view of vertical collaborating entities ··· 8
Figure 2.2 Collaborative partners linked by the Internet ··· 11
Figure 2.3 Collaborative production planning system structure ··· 13
Figure 3.1 ABC model ··· 16
Figure 3.2 An example of activity transition diagram ··· 19
Figure 3.3 A generalized ABC Pr/Tr Net model ··· 21
Figure 3.4 WT and UT of an activity ··· 24
Figure 4.1 Transformation of material in process ··· 29
Figure 4.2 Assemble and testing process ··· 30
Figure 4.3 Cost drivers and activity costs for assembly and testing firm ··· 34
Figure 4.4 Activity transition diagram ··· 35
Figure 4.5 The ABC Pr/Tr Net model ··· 37
Figure 5.1 (a) Data structure for a resource token ··· 42
Figure 5.1 (b) Data structure for an order token ··· 42
Figure 5.1 (c) Data structure for rule ··· 42
Figure 5.1 (d) Data structure for schedule ··· 42
Figure 5.2 (a) A reasoning tree for goal 1 ··· 49
Figure 5.2 (b) A reasoning tree for goal 2 ··· 50
Figure 5.3 The system structure of ABC/CPPS ··· 51
Figure 5.4 The GUI of the ABC/CPPS ··· 52
Figure 5.5 Client/Server web site structure ··· 53
Figure 6.1 The scenario ··· 54
Figure 6.2 The reasoning tree for example case ··· 54
Figure D.1 Collaborative production planning and the data exchange ··· 71
Figure D.2 DTD example ··· 73
Figure D.3 XML document example ··· 73
Figure D.4 Display by IE web browser for Figure 6.3 ··· 74
Figure D.5 Abstract tree structure—DTD for collaborative schedule ··· 76
Figure D.6 The DTD for collaborative schedule ··· 77
Figure D.7 The excepted display of XML document ··· 77
Figure D.8 XML document structure ··· 78
Figure D.9 Main functions of program ··· 78
Figure D.10 Collaborative schedule (XML format) ··· 79

LIST OF TABLE CONTENTS

Page

Table 4.1 Define and analyze the indirect manufacturing activities ························31

Table 4.2 Define and analyze the activities/resources for assembly firm ················31

Table 4.3 Define and analyze the activities/resources for testing firm ····················32

Table 5.1 The static transition logic rules for ABC Pr/Tr net model ·························45

Table 5.2 The dynamic transition logic rules for ABC Pr/Tr net model ··················46

Table 6.1(a) Basic data for assembly firm ASE in example case ·····························56

Table 6.1(b) Basic data for testing firm ASET in example case ·····························57

Table 6.1(c) Basic data for testing firm WTAE in example case ·····························58

CHAPTER 1

INTRODUCTION

1.1 Research Motivations

In conventional semiconductor backend manufacturing, firms operate independently. But nowadays, the market confronts an environment of low-profit margin. Therefore, it is hard for an independent firm, especially a small one, to survive. It is also noted, semiconductor industry encounters a cyclic market trend, over investment in capital will incur fatal risk for an enterprise during the low season. However, strategic alliance with other partners, or integrating their own internal business units to form a Virtual Enterprise (VE), in order to enhance competitive advantages and share the market risk/opportunity, seems promising. Recently, the Internet technology evolves rapidly, and new concepts to make the best use of this technology have emerged concurrently over the past few years. One of the popular concepts is using the Internet to create a collaborative environment that is characterized by network cooperation, autonomous self-control and transparent communication. And, it seems collaborative operation is capable of providing more competitive advantages based on the following reasons.

- Take the advantages from the Internet and VE.
- Each firm can focus on its core competency.
- To enhance the utilization of resources by collaborative operation.
- To share the market risk/opportunity.
- Make the enterprise more agile and flexible by downsizing and dedicate to specific competitive process.

Two kinds of measures are used to evaluate system performance, the financial and the non-financial measure. Most past production research on semiconductor industry focused on a single firm's planning/improvement using non-financial measures, such as maximum throughput, maximum utilization of workstations, minimum production cycle time, least tardiness or maximum wafer movements (Wein 1988, Glassy and Resende 1988, Spearman *et al.* 1990, Uzsoy *et al.* 1992, Johri 1993, Liao *et al.* 1996). But these measures do not closely or directly link to an enterprise's overall financial vision. Fisher (1992) stated that, one of the key difficulties of the non-financial system was its inability to dollarize the amount of improvement in the non-financial measurements, i.e., the connection between improvements in the non-financial measures with profits was unclear. According to the research conducted by Laitinen (2002), both financial and non-financial measures are important, but small technology firms appear to emphasize the importance of company-level profitability and other financial performance factors.

A pyramid model was proposed by Lynch and Cross (1991) as shown in Figure1, they claimed that it is useful for describing how objectives are communicated down to the low levels and how measures can be rolled up to various higher levels in the enterprise. A production planning manager should not only to ascertain the breakdown of objectives for each department but also should know the consequences of the planned activities rolled up to the enterprise's overall objectives before a financial report is prepared. Therefore, in addition to the non-financial measures commonly used in the semiconductor industry, this study intends to supplement a financial performance measure by applying activity-based costing (ABC) for collaborative production planning (CPP).

The ABC system was first introduced in 1971. Basically, it is a concept that product consumes activities and activities consume resources therefore the product cost can be derived. As suggested by Salafatinos (1996), ABC systems can be adapted to provide profitability information for enterprise activities. It represents a kind of paradigm for evaluating the economic consequence of production planning decisions. Recently, ABC has been expanded to the management area in form of ABM (Activity Based Management). It is an area of decision support that can directly connect to operational planning in order to provide useful financial information.

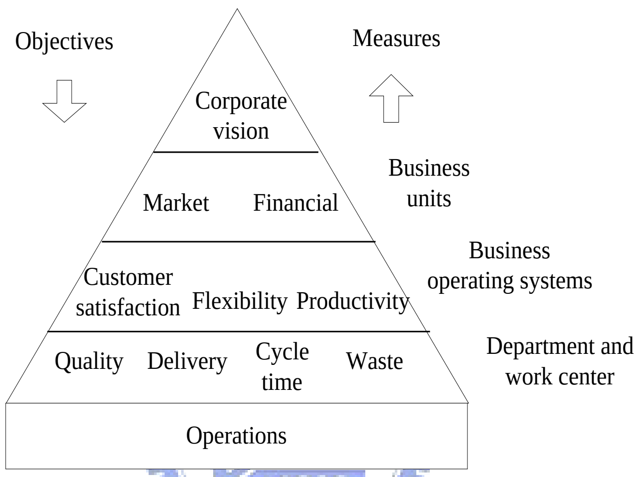


Figure1.1 The performance pyramid

Though ABC has demonstrated that it is a suitable financial measure, and past research generally used spreadsheets to implement ABC in estimating product cost, but as Pirttilan and Hautaniemi (1995) pointed out that the data needed for ABC has to be handled with computers, small models can be created with spreadsheet programs, but large systems need more effective methods and the programming might be problematic. It also was discovered that most estimations of manufacturing costs used only static method at the early stage of design, without considering the actual constraints and competition of resources in the dynamic production environment. Moreover, it should be noted, that even though activities with the same cost driver will result in different cost values impacted by the specific type of resources actually consumed. A static procedure may not treat the actual cost appropriately. Therefore, applying a computer-based tool that incorporates the ABC model along with a simulation methodology seems applicable to effectively address these problems.

1.2 Research Objectives

This study aims to develop an Activity-Based Costing/Collaborative Production Planning System (ABC/CPPS), to help the production planner to estimate the financial consequences of production planning and create the collaborative schedule(s) for each partner to fulfill in a semiconductor backend turnkey (combined IC assembly and testing) operational service environment.

Based on the ABC theory, an ABC Pr/Tr Net simulation model can be developed and implemented systematically. The financial measure, profit, for the VE can be estimated and the collaborative schedule(s) for each partner to cooperate can be created. An example is illustrated to demonstrate the potential possibility for application.

1.3 Scope and Assumptions

As shown in Figure 1.2, the entire supply chain of the semiconductor industry is linked by IC design, fabrication/probing, assembly and testing firms. Basically, the semiconductor backend turnkey service is composed of assembly and testing operations, some but not all semiconductor backend enterprises own both assembly and testing firms. Due to the heavy investment required, most of past research has focused on front-end (fabrication) instead of backend (assembly and testing) (Liu *et al.* 1997). Studies about production planning considered turnkey operational service had appeared infrequently. Even though there were some studies related to the testing operations, most of them simplified the complexity of the manufacturing process on the testing floor (Liu *et al.* 1997, Yang and Chang 1998, Freed and Leachman 1999). Therefore, scope of this study is defined as turnkey service, which includes both assembly and testing shop floor operations.

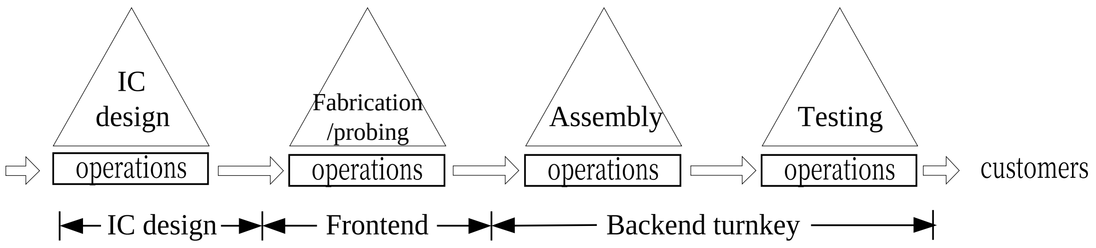


Figure 1.2 The entire supply chain of semiconductor industry

About the whole process of product delivery, four cycles are identified; the product design/development cycle, the manufacturing cycle, the sales cycle and the marketing cycle. But it was found most research about collaboration focused on the design/development cycle, little attention paid to other areas. Therefore, this study devotes to collaborative planning for manufacturing. And the vertical collaborative operational scenario (enterprise links the upstream or downstream heterogeneous business partners to provide a favorable turnkey service, such as combined assembly and testing) is focused in this study. Moreover, the one assembly-to-many testing firms operations environment is illustrated.

In this study, profit calculation depends on cost data; therefore, cost of each operation for each plant is available by means of ABC methodology and cost of each operation for each plant might be different are assumed in this study. Sale price for each order is assumed to be known too.

1.4 Organization

In this study, the definition, types and key issues for collaborative operation are examined first, and then a collaborative production planning view and a system structure which is composed of ABC data and ABC Predicate/Transition Net (Pr/Tr Net) model is proposed in chapter 2. Methodology for the development of an ABC/CPPS is proposed in chapter 3. Based on the proposed methodology, the ABC data and the ABC Pr/Tr Net simulation model used to simulate the resources consuming process are developed in chapter 4. The detail implementations of the ABC Pr/Tr Net, such as data structure design and logic programming are conducted in chapter 5. Moreover, an example case is illustrated in chapter 6 to demonstrate the potential application capability. Finally, conclusion and future research possibilities are presented in chapter 7.

CHAPTER 2

COLLABORATIVE PRODUCTION PLANNING FOR SEMICONDUCTOR BACKEND ENTERPRISE

2.1 Definition and Types of Collaborative Operation

The "collaborative operation" is defined as enterprises, with either homogeneous or heterogeneous characteristic, are organized together to cooperate and share the collaborative resources released from each collaborative member under the Internet environment, in order to achieve an operational goal. The scope of collaborative operation could either integrate the whole activities ranging from booking the orders through design, production and marketing, or focus on a individual segment. Usually, a VE (or physical enterprise) may be formed to coordinate the whole operation through the Internet. Among past researches, it was found most of them focused on collaborative product design/development cycle, lack of research investigated into the collaborative production.

Two kinds of scenarios are available for collaboration, the horizontal collaborative scenario and the vertical collaborative scenario. Horizontal collaborative scenario is defined as homogeneous business partners allied together in order to provide an operational service on a cooperative basis by integrating and sharing the collaborative resources from each others. In vertical collaborative scenario, enterprise links the upstream or downstream heterogeneous business partners to provide a favorable turnkey service (such as combined assembly and testing). If applied properly, both kinds of collaborative scenarios can improve the competitive advantages for the VE. In this study, focus is placed on vertical collaborative scenario applying to semiconductor turnkey service in a one assembly-to-many testing operational firms' environment. By taking the advantages of the Internet, visibility on available resources released by each cooperative member for collaborative operation is easily enhanced. This facilitates the collaborative operation.

Figure 2.1 depicts a conceptual view of vertical collaborating entities. The dotted triangle across two firms represents a VE (or physical enterprise) with an overall enterprise view that considers integrating internal business units or forming a strategic alliances with other trade partners by applying CPP to achieve an overall consensus goal, which benefits going to the integrated backend enterprise and/or to the individual firms simultaneously. Under this centralized collaborative planning scheme, resources owned by each firm will be co-planned together to configure the best arrangement for customer orders (COs).

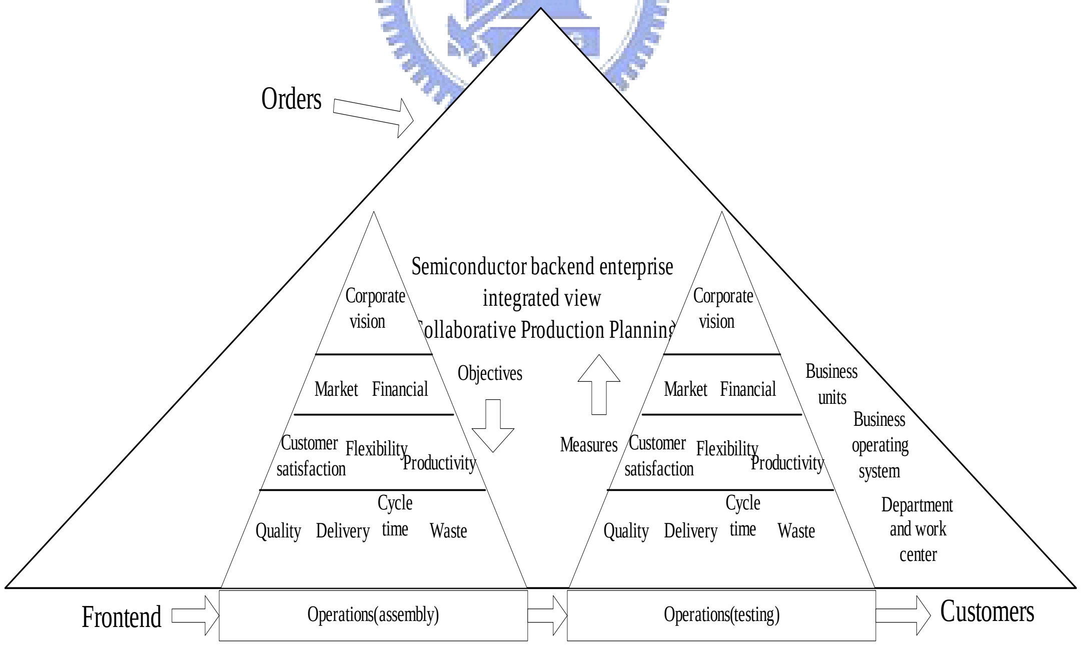


Figure 2.1 Conceptual view of vertical collaborating entities

Using the advantage of the Internet, the CPPS system can be designed as a web-based system, which is easily accessed by each member (Huang and Mak 1999). The COs processing procedure will have little if any change, COs previously were accepted and confirmed by each separate firm now with the change to a single, integrated window of the integrated VE. Profit is important in such a cooperative initiative. Without profit, a firm cannot sustain its operations and thus will negatively impact the cooperative relationship.

2.2 The Key Issues for Vertical Collaborative Operation

Concept close to collaborative operation had been appeared in the form of HMS (Holonic Manufacturing System). HMS is an approach of theoretical framework for autonomous and decentralized manufacturing based on the classical holonic theory introduced by Keostler (1989). In 2000, Mezgar *et al.* (2000) ever proposed a new co-operative manufacturing network model for coordinating the production of small-and medium-sized enterprise (SMEs), based on the holonic paradigm. And Huang *et al.* (2002) proposed a framework for virtual enterprise control with the holonic-manufacturing paradigm.

In the vertical collaborative scenario, heterogeneous business partners from somewhere are allied and organized as a VE. Therefore some characteristics, indeed exist among the collaborative partners, are investigated and discussed as follows.

- Autonomy: Each member in the collaborative environment is an autonomous entity with capability to create and execute its own plan and/or strategies.
- Cooperation: Allied members cooperate together, in order to toward a global production goal; usually, collaborative schedules that are acceptable and executable by all the collaborative members are created.
- Homogeneous process capability: For a specific process to be executable

among all allied members, the process capability of the collaborative members must be homogeneous and capable to meet the quality level required by customer. Since the assembly yield of most assembly firms in semiconductor industry can reach above 99%, therefore, it is assumed that the process capability is homogeneous.

- Collaborative operation: A collaborative operation is defined as an operation that is exchangeable and executable among the allied members, i.e. any member in the collaborative environment is capable in performing this operation, and thus a collaborative operation can be scheduled and assigned to any member.
- Process segment: Though individual operation can be classified into collaborative operation or non-collaborative operation, but collaborative operation may be limited by certain specific constraints, such as some operations must be performed in a clean room. This constraint makes these operations to form a process segment and to be preferably assigned and continuous executed by a single specific member under the same clean room environment.
- Collaborative resources: Collaborative resources are resources (such as machines) with available time released and declared by collaborative members. Part or all resources of an allied member can be declared as collaborative resources, which are used to plan the collaborative schedules for all members to execute cooperatively.
- Geographical location limit: One of the conditions to make collaborative production feasible is the location of each collaborative manufacturing partner is not geographically far away.
- Internet support: Beside the collaborative operation takes the advantages of

the Internet, moreover, the web-based client and server architecture is found to be attractive for collaborative operation. As shown by Figure 2.2, the server and client PC use the HTTP (Hyper Text Transfer Protocol) to exchange the HTML (Hyper Text Markup Language). Based on the web-server architecture and the Internet infrastructure, collaborative members can be networked through the Internet to provide a collaborative scheduling/manufacturing environment.

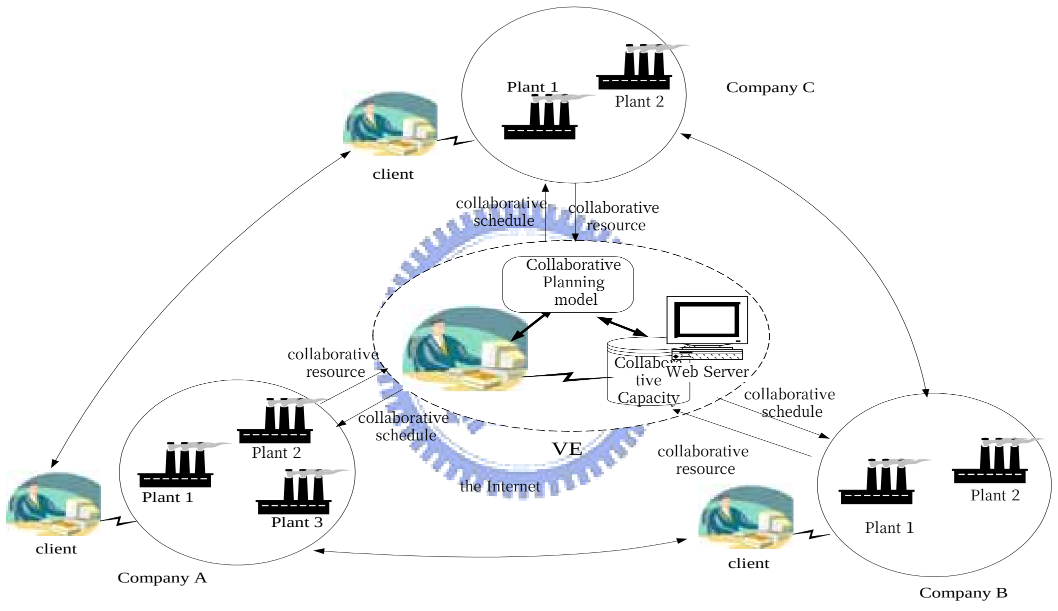


Figure 2.2 Collaborative partners linked by the Internet

- Collaborative planning platform: To facilitate the implementation of collaborative operation, a collaborative planning model is required. In practice, a collaborative planning model is built to serve as the planning engine based on available collaborative resources. After the collaborative manufacturing schedule is planned, the results may be published to web server and directly assessable by browser from the client, or transferred to each member by using XML (eXtensible Markup Language).

2.3 A Collaborative Production Planning System Structure

Three factors significantly impact the performance of production system are identified; the release rule, the dispatching rules and the production policies. Release policy states about the mix and release of orders to start the manufacturing process at certain point of time, dispatching rules are used for deciding the next lot to be processed or which machine to be assigned to process the lot on the shop floor, and the production policies relate to other strategies (or decisions) which might impact the production plan (such as outsourcing policy). It is reasonable to assume that system performance (profit) will be positively impacted by these two factors in an integrated collaborative production endeavor. Most semiconductor firms commonly use Earliest Due Date (EDD) as release and dispatching rules to schedule and meet customer requirements (Kim *et al.* 1998). But how performance of EDD directly links to the enterprise's profit is usually an unknown.

About the dispatching strategies, three kinds of rules, the sequential, the priority and the variable priority rule were identified by Agliare *et al.* (1995). Moreover, another classification scheme was proposed by Rajendram and Holthaus (1997), who claimed dispatching rules can be classified into five categories: (1) rules involving processing time, (2) rules involving due-date, (3) simple rule involving neither processing time nor due-date, (4) rules involving shop floor conditions and (5) rules involving two or more of the first four classes. Although, various kinds of dispatching rules have been proposed, but after many past exhausting researches devoting to this topic, one unchangeable conclusion is that no single rule can be found to perform well for all important measures.

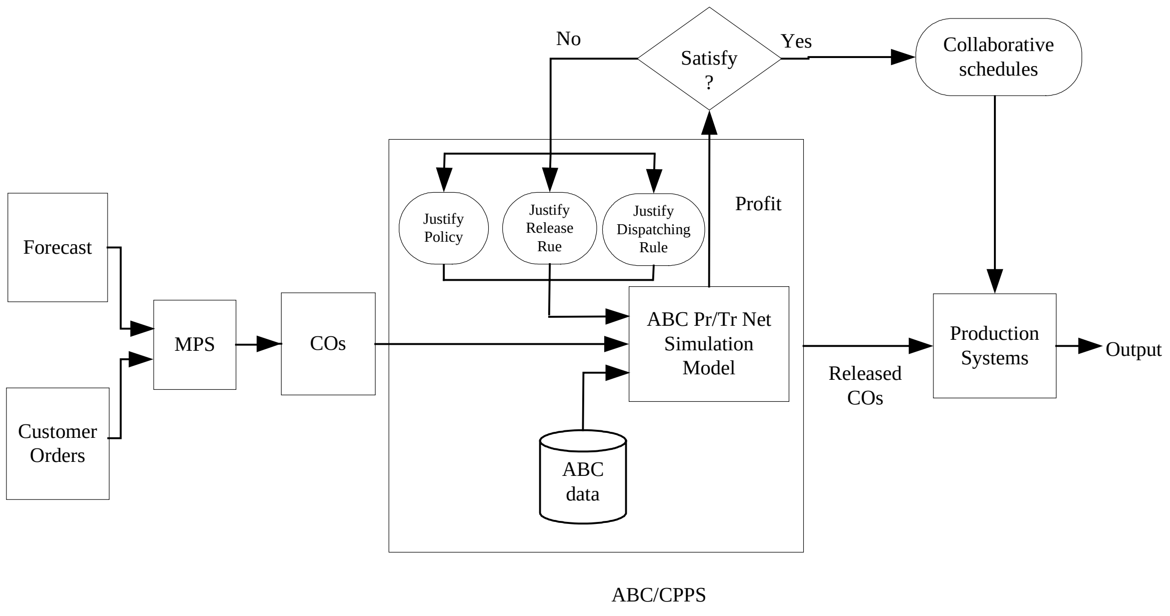


Figure 2.3 Collaborative production planning system structure

A system structure of collaborative production planning composed of ABC data and Pr/Tr Net model to simulate the resource consuming process for semiconductor backend turnkey operational service is proposed as shown in Figure 2.3. In the system structure, COs are selected by release policy from the Master Production Schedule (MPS). Before the release to the production line, COs are transformed to the planned manufacturing orders (MOs). The activity-based costing collaborative production planning system (ABC/CPPS) is used to simulate and roll up all the activity costs by the consuming resources for the released COs. Profit is calculated and the collaborative schedule for those released COs is reported to the production planner. If the expected outcome is acceptable then the planned MOs are released to production line for manufacturing. Otherwise, either justifies the release policy, planning policy or dispatching rule to have a different collaborative schedule used on the shop floor for each partner are investigated.

CHAPTER 3

METHODOLOGY

3.1 The Existing Decision Models

Several existing decision models for problem solving have been developed. The power and limits of these models are listed below.

(1) Static Allocation Model: is a static and simple model, it ignores all the dynamics, interactions, and various measures of performance, though the static allocation model is easy to implement, it can be too inaccurate and seriously overestimates systems' performance.

(2) Queuing Network Model: the basic theory of queuing network was developed by Jackson (1957), and later on extended by Gordon and Newell (1967) and, Buzen (1973). This kind of model accounts for the dynamics, interactions, and uncertainties in the system. But a disadvantage of the queuing network model is that, a set of restrictive assumptions (e.g. exponential processing times, infinite queues) is often required.

(3) Simulation Model: can provide an accurate picture of system performance, but it takes a long time for a model building and data input.

(4) Perturbation Analysis: Detailed behavior of the system is observed, either through simulation or from the actual system in process for one set of decision parameters. By doing some minor additional calculations while the system is being observed, Perturbation Analysis can predicate the system behavior if decision parameters are changed. The main disadvantage of this model is that it cannot accurately predicate the effects of large changes in decisions.

(5) Petri net Model: is quite appropriate for modeling dynamic systems. In addition, it is graphical, readable and easy to understand.

This study shows that system performance depends on the characteristic of each unique system, and giving the complexities of a manufacturing system, it is clear that analytical, queuing network and perturbation models are not easily adaptable for modeling a unique dynamic system. Therefore, a simulation-based model combined with Petri net could be appropriate for modeling a dynamic manufacturing system. And in recent years, the use of simulation-based finite capacity planning and scheduling software has increased dramatically in the semiconductor industry.

3.2 The ABC/M Model

ABC was introduced by George and Staubus in 1971. As shown in Figure 3.1, the ABC model describes product (cost object) consumes activities and activities consume resources. Therefore, from the two-stage structure, product cost can be derived. Some researchers (Pirtila and Hautaniemi 1995, Tsai 1996) pointed out that ABC offer more accurate product cost than the traditional cost system by using cost drivers to trace the costs of activities consumed by product/order.

Activities in ABC model are categorized into unit-level, batch-level, product-level and facility-level proposed by Cooper (1990). Unit-level cost is defined as inputs increase in proportion to the number of units processed, such as number of wafers to be grinded, numbers of dies to be wire bonded, number of dies to be tested, etc. Most of the activities occurred on the shop floor can be attributed to unit-level cost. Batch-level cost assumes that inputs vary in

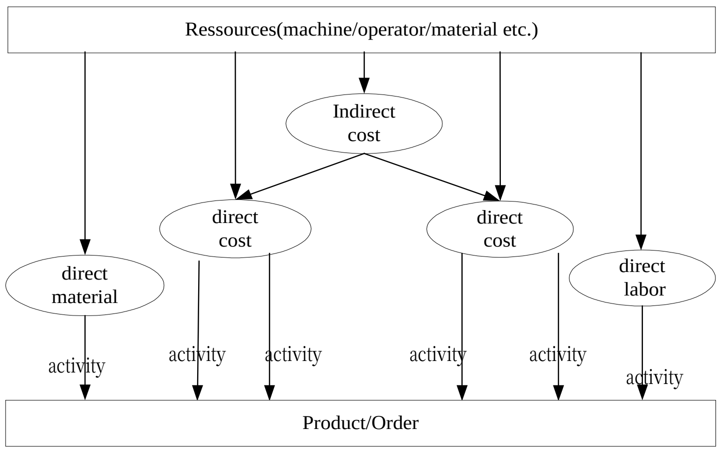


Figure 3.1 ABC Model

proportion to the number of batches processed, such as orders to be processed, set-up required whenever a batch of product is to be manufactured. Product-level cost assumes that inputs are necessary to support the manufacturing of each different type of product, such as inventory holding for all completed products. A facility-level cost is those costs related to sustaining a facility's general manufacturing process, such as a general administrative cost. Except the facility-level cost is more difficult to estimate (Cooper 1990, Foster and Gupta 1990, Ong 1995), the other three costs can be directly attributed to individual cost object.

Using another point of view, activity costs are classified into direct material, direct labor and some overhead costs. The core elements and types of costs in the ABC model are defined below:

- Cost Object: the object that consumes activities, such as product or order, the costs of activities are calculated and rolled up to the cost object.
- Cost driver: the factors incur costs for a specific activity, such as the process time, the quantity of product or the quantity of material consumed.
- Direct material: the cost of purchased material, such as lead frame, gold wire,

etc. directly consumed in the production line and can be attributed to the cost object. The direct material can use standard material cost data.

- Direct labor: the cost is determined by multiplying the standard time with average mean hour salary rate in the production center, such as a 'set up' by the operator.
- Overhead: In tradition accounting, overhead costs include variable and fixed cost. Variable overhead cost includes repair, maintenance, supplies (tool service, QC) and office-depreciation. Repair overhead cost occurred stochastically on the shop floor. Fixed overhead cost includes salaries for staff personnel, rental for a building or a machine, utilities, water, indirect labor, planning, preparing, production administration and other remains. According to the ABC model, indirect activities are reconceptualized to direct activities and assigned to cost object (product/lots) directly as possible (Armstrong 2002).

Among these costs, the allocation of overhead costs as a percentage of direct labor cost and/or machine usage by traditional accounting may not accurately reflect the actual product cost, and the more complex, low-volume and small batch products tend to be underestimated by traditional accounting methods (Cooper and Kaplan 1988, Dhavale 1990); therefore, many companies have improved their cost accounting by developing an ABC system. Lea's research (2002) found that ABC provides higher profit, lower inventory and better overall service across multiple manufacturing systems than a throughput accounting. Recently, ABC has been expanded to the management area in form of ABM (Activity–Based Management), it is an area of decision support that can directly connect to operational planning in order to provide useful financial information.

In contrast to ABC, the traditional accounting system is designed mostly to meet financial accounting purposes and therefore focuses on creating balance sheet and an income statement. ABC appeared as a more powerful tool in decision-making area.

About the development of ABC, a seven-step methodology for designing an ABC system had been studied by Pirttilan and Hautaniemi (1995). The steps proposed in that paper are: (1) Scope of interest, (2) Documenting the manufacturing process, (3) Define activities/resources, (4) Analyze activities/resou, (5) Select cost drivers, (6) Activity costs, (7) Cost object costs.

3.3 The ABC/CPPS

Though Pirttilan and Hautaniemi (1995) provided a seven-step methodology, it was discovered that most past implementation of ABC used only static method at the early stage of product design, without considering the actual constraints and competition of resources in the dynamic production environment. Moreover, it was noted, that even though activities with the same cost driver will result in different cost values impacted by the specific type of resources actually consumed in the shop floor. Therefore, a static procedure may not treat the actual cost appropriately, and applying a computer-based tool that incorporates the ABC model along with a simulation methodology seems applicable to effectively address these problems.

As shown in Figure 2.3, the dotted rectangle with two core components, the ABC data and the ABC Pr/Tr Net simulation model, is defined as the scope of the ABC/CPPS. Therefore, a two-stage approach with detail steps is proposed to develop the ABC/CPPS.

Stage 1: Analyze the ABC data

1. Define the domain of application: firstly, we should define the scope we are interested to apply, and the collaborative entities must be identified.
2. Understand the processes of the application domain: according to the domain of the application, the operations processes for the collaborative entities must be investigated.
3. Analyze the activities/resources from the processes: extract activities and analyze the resources used for each activity from the processes, and then assign an activity code to each activity in order to be traced by the simulation model in stage 2.
4. Analyze the cost drivers and decide the cost object: after extracting the activities in the previous step, cost drivers and activity cost for each activity are analyzed, and then a cost object must be decided.

Stage 2: Develop the ABC Pr/Tr Net simulation model

1. Analyze the activity transition: in order to simulate the sequence of activities for rolling up cost, an activity transition diagram is proposed as a tool in this step. As shown in Figure 3.2, an example is illustrated for two collaborating entities with a sequential activity transition, where activities a_1 to a_p belong to entity A and activities a_{p+1} to a_q belong to entity B. Therefore, a universal set $X=\{a_1, a_2, \ldots, a_p, a_{p+1}, a_{p+2}, \ldots, a_q\}$ can be derived.

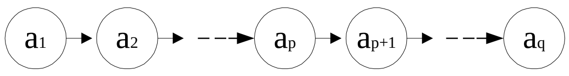


Figure 3.2 An example of activity transition diagram

2. Group the activities into activity sets: analyze the activities in the universal set X and respectively categorize them into sets Y and Z, each contains subsets. Each subset in Y is used to control the trigger of a specific transition and transit lots to use resource(s). On the other hand, each subset in Z is used to control the trigger of a specific transition and transit lots to end of resource(s) usage. The purpose of grouping the activities into different subsets by specific characteristic is that we can control similar activities to be transited by a specific transition, which includes specific treatment for this group. Set X, Y and Z will be used in next step and are defined as follows:

   X: X is the universal set consists of all activities; $X=\{a_1,a_2,.....a_p,a_{p+1},a_{p+2},.....a_q\}$.

   Y: Y is a set constructed by subsets Y_i ; $Y=\{Y_i\ ;\ i=1,m\}$, Y_i is a subset consists of activities with same characteristic to trigger a specific transition which transits tokens (orders) to use resource(s). The definition of characteristic depends on application requirement or concern. Such as, activities that require resource(s) supply might be defined as a subset, or activities require setup operation might be grouped into one subset, etc. Note that exclusive relationship $Y_1 \cup Y_2 \cup \ldots \cup Y_m = X$ hold.

   Z: Z is a set constructed by subsets Z_j ; $Z=\{Z_j\ ;\ j=1,n\}$, Zj is a subset consists of activities with the same characteristic to trigger a specific transition which transits tokens (orders) end of resource(s) usage. The definition of characteristic depends on application requirement or concern. Such as, activities cause lots requiring a merge process after resource usage might be grouped into one subset, etc. Again, note that exclusive relationship $Z_1 \cup Z_2 \cup \ldots \cup Z_n = X$ hold.

3. Develop the simulation model: A generalized ABC Pr/Tr Net with a loop structure is proposed as shown in Figure 3.3 to simulate the dynamic and iterative consuming process.

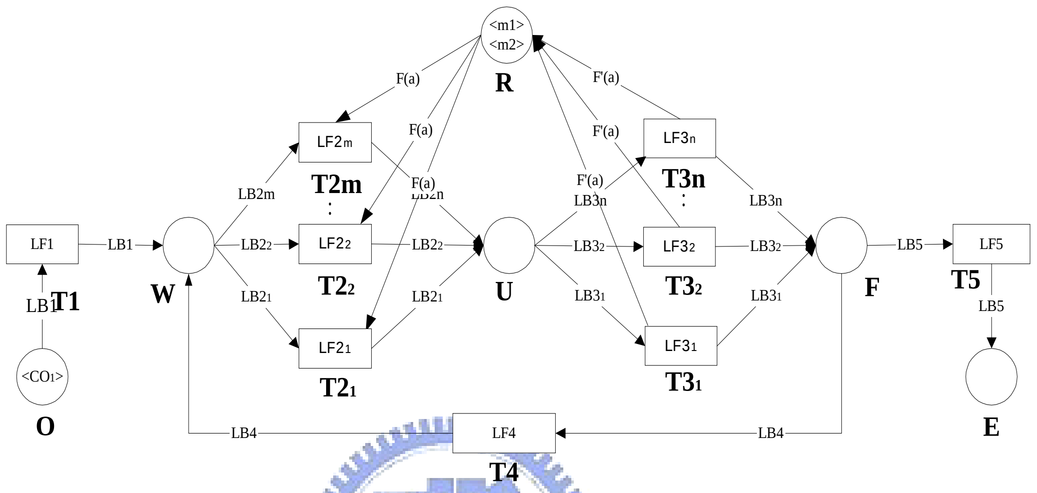


Figure 3.3 A generalized ABC Pr/Tr Net model

Pr/Tr Net is a high-level petri net, which possess higher abstraction and aggregation properties for modeling. Basically, Pr/Tr Net is a directed graph (P, T, A) where P is the set of predicates ('first-order' facts), T is the set of transitions, A is the set of arcs and some other components, logical formulas, labels are used to constitute the Pr/Tr Net. For the detail definition of Petri net and Pr/Tr Net please refers to Genrich and Lautenbach (1986), Murata (1989) and Lee *et al.* (1994). Components of the proposed ABC Pr/Tr Net model are defined below:

- Predicate: P={O, W, R, U, F, E} are predicates to state the facts.

  O: There is an Order.

  W: Want to use resource; an order is ready to consume the resource(s).

U: Using resource; describe the consuming process, and rolling up the cost.

R: There is a resource.

F: Finish using resource; the consuming process is complete.

E: Exit system; all activities for an order are complete.

- Transition: T={T1, T2, T3, T4, T5} are transitions to transit predicates.

  T1: Transit lots from O (There is an order) to W (Want to use resource) predicate.

  T2: T2={$T2_i$; $i=1,m$} transit lots from W predicate to U (Using resource) predicate.

  T3: T3={$T3_j$; $j=1,n$} transit lots from U predicate to F (Finish using resource) predicate.

  T4: Transit lots from F predicate to W predicate (Want to use).

  T5: Transit lots from F predicate to E (Exit system) predicate.

- Logical formula: A formula expressed by logic syntax inscribed in a transition. Such as $a \in X$ is a logical formula with vale of 'true' only if a, belongs to set X. A set of formulas LF= {LF1, $LF2_{i=1,m}$, $LF3_{j=1,n}$, LF4, LF5}, is used in the net, contents for each logical formula depend on application. LF1 is the logical formula which states the condition to release an order into production, $LF2_{i=1,m}$ are logical formulas used to control the triggers of the transitions $T2_{i=1,m}$ and transit a lot to U predicate for resource(s) consuming, $LF3_{j=1,n}$ are the logical formulas used to control the triggers of transitions $T3_{j=1,n.}$ Similarly, LF4 is a logical formula used to control the trigger of transition T4 and LF5 is used to judge the trigger of transition T5.

- Label: are labels of formal sum for some arcs in the Pr/Tr Net, for example, <B> is a label, and <A, T> is another label, a formal sum may be expressed by <B> + <A, T>. In the label, B, A and T are attributes (like a variable) used in these labels. A set of labels LB={LB1, LB2$_{i=1,m}$, LB3$_{j=1,n}$, LB4, LB5} is used in the net, contents for each label depend on application.
- Firable: A transition is defined 'firable' whenever its preconditions and logical formula are satisfied. For example, T1 is firable, if there is a token O<CO$_1$> exist, resources determined by functor F(a) are available (for activity a) and logical formula LF1 is satisfied. After T1 fired, order token O<CO$_1$> will be transited to token W<CO$_1$> (CO$_1$ want to use).

This paragraph describes the details of the running of the ABC Pr/Tr Net model. First, COs reside at O predicate, all resources such as operators, machines, etc. reside at R predicate. The resources constitute the production capacity of the system. Moreover, the COs and the resources initially reside at O and R predicates respectively form the initial markings of the net. When the model begins to run, customer orders (or 'tokens') selected by the release policy are transited (by transition T1) to predicate W with the initial activity code (a=1). At W, if a lot is selected by a dispatching heuristic rule (such as EDD) and acquires all the required resources allocated by functor F(a), then transition T2$_i$ will be fired (depends on LF2$_i$ is satisfied) and the lot is transited to U predicate for operation. After completing this operation, T3$_j$ then will be fired (depends on LF3$_j$ is satisfied) and resource(s) will be returned to R predicate by functor f'(a). Finally, the lot is checked at F predicate by LF5 to make sure all activities are completed, then the lot will be transited to E (Exit) predicate, otherwise the lot will be transited back to W predicate and recycled again for next activity. When all lots reach E predicate (all lots are finished), the model then stops and concludes the profit.

4. Profit calculation

In previous step, the ABC Pr/Tr Net is used to simulate the resources consuming. In essence, there are two kinds of predicates in the ABC Pr/Tr Net simulation model, ‘activity’ predicates (U) and ‘state’ predicates (W, R, O, F, E). Only at activity predicates U, lots may hold and consume resource(s). Other ‘state’ predicates only show a state of the lots. As illustrated by Figure 3.4, the WT (Waiting Time) and UT (Using Time) of each activity can be simulated and therefore the cost can be estimated according to the activity cost drivers. Finally, the Total Manufacturing Cost (TMC) can be derived when all orders are completed.

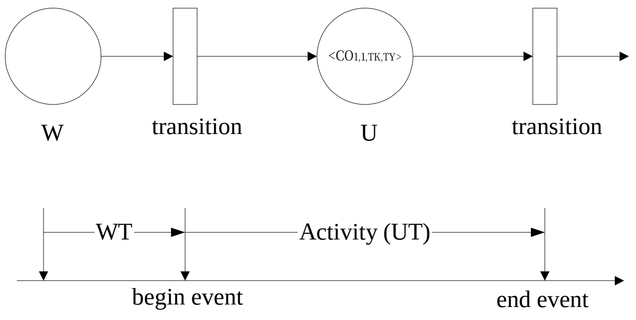


Figure 3.4 WT and UT of an activity

Manufacturing Net Profit (MNP) of the released COs is calculated by subtracting the Total Manufacturing Cost (TMC) from Total Sales Revenue (TSR). TSR is derived by multiplying sales order quantity by unit prices per piece for all the released orders. The manufacturing cost considered in this study includes direct labor cost, direct material cost, and some overhead costs driven directly by the cost drivers, which were defined in section 3.2. Some overhead costs, such as repair that occurred stochastically and other overhead costs (like water, electric, insurance, office-depreciation, maintenance, administrative cost and other remains), which are minor or not impacted by the release policy and/or dispatching rules, are neglected and not included.

CHAPTER 4

DEVELOPMENT OF ABC/CPPS

A case is given in this chapter to illustrate the application of the two-stage approach for ABC/CPPS development proposed in previous chapter. Section 4.1 corresponds to the development on stage 1, and section 4.2 corresponds to the development on stage 2.

4.1 Analyze the ABC Data

4.1.1 Define the Domain of Application

This case is based on W Corporation, an enterprise that owns both assembly and testing business units and focus on producing DRAM, located in Taiwan. Three kinds of product types, assembly only, testing only and turnkey products are considered in this case. The process needed for each order depends on product type. Assembly only product type needs only assembly processing, both testing only and turnkey product must perform the testing process.

4.1.2 Understand the Process of Semiconductor Backend Operation

An Oracle ERP (Enterprise Resource Planning) system initiates order processing (O/P), manufacturing orders (MOs) are created from COs. Before the MOs are released, the material check (M/C) and program/tool check (PT/C) activities are performed to make sure of the production feasibility. After order release (O/R), an inventory retrial (I/R) is conducted to access materials and wafers for releasing to the production line.

As shown in Figure 4.1 (illustrates the transformation of material in process) and Figure 4.2 (gives the details of the process), the assembly process is somewhat of a flow type process. The entire assembly manufacturing process can be divided into pre-assembly and assembly processes. The pre-assembly process consists of W/G (Wafer Grinding) and W/S (Wafer Saw) operations, the others belong to the assembly process. After dies have been sawed out during the pre-assembly process, the batch (lot) is moved to the assembly process. A series of operations, including D/B (Die Bonding), W/B (Wire Bonding), M/D (Molding), D/C (Dambar Cutting), S/P (Solder Plating) and T/F (Trim & Forming) are performed to assemble and package the dies. The W/B operation usually becomes the bottleneck during the assembly process. Therefore, after the D/B operation, the MOs will be split into sub-lots for production efficiency. And finally, a V/I (Visual Inspection) operation is performed to make sure of the appearance quality, if the product type for the lot is assembly only, then lots for the same customer will be combined, packed (A/P) and shipped out (S/O) to the customer, otherwise they will be transferred to the testing firm.

On the testing shop floor, firstly the dies will go to burn-in (B/I) operation, after burn-in, dies go to low temperature testing (FT1), according to the testing results, dies will then be classified to pass bin, fail bin, etc. Dies might need to rework FT1 if high defect results come out. After FT1, dies go to high temperature testing (FT2), when this operation is finished, speed test (FT3) is conducted and bin classes are set for classifying the dies (split). After speed test, dies of each bin are marked (M/K). An Engineering Quality Assurance (EQA) gate is set to determine pass or fail after the marking operation. If pass, then Lead scan (L/S) operation is performed to make sure all leads of dies in the same level. After lead scan, a visual inspection (VI) is conducted to check the dies appearance. When no problem occurs, baking (B/K) operation is performed to dry the dies.

After this operation, if package type for this lot is ‘tape and reel’ then the dies will move to tape and reel operation (T&R), otherwise lot will forwarded to packing operation (P/K) directly. Within 6 hours after the baking operation, dies must be completely packed (P/K). A Final Quality Assurance (FQA) is performed to make sure of the outgoing quality.

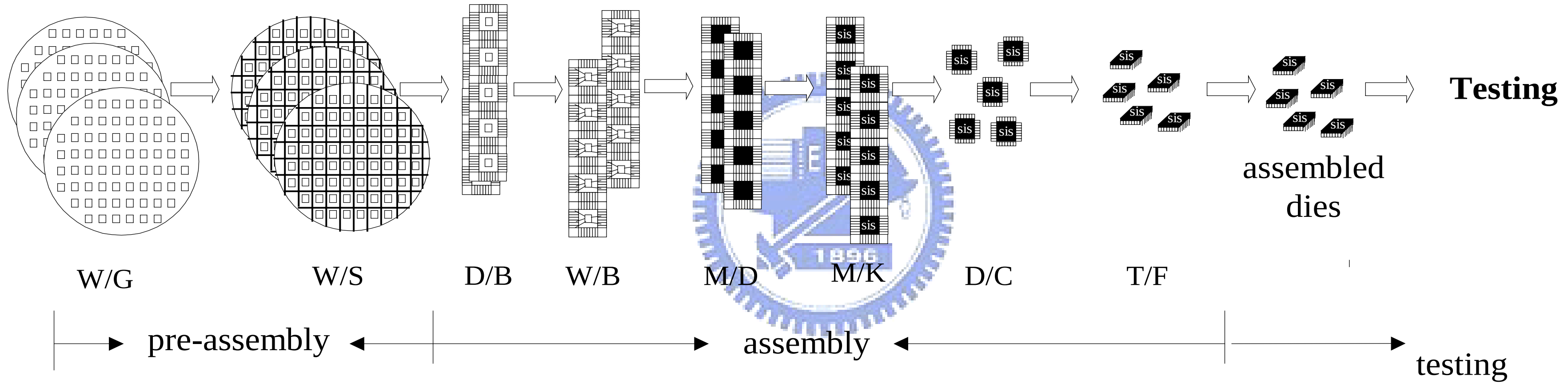


Figure 4.1 Transformation of material in process

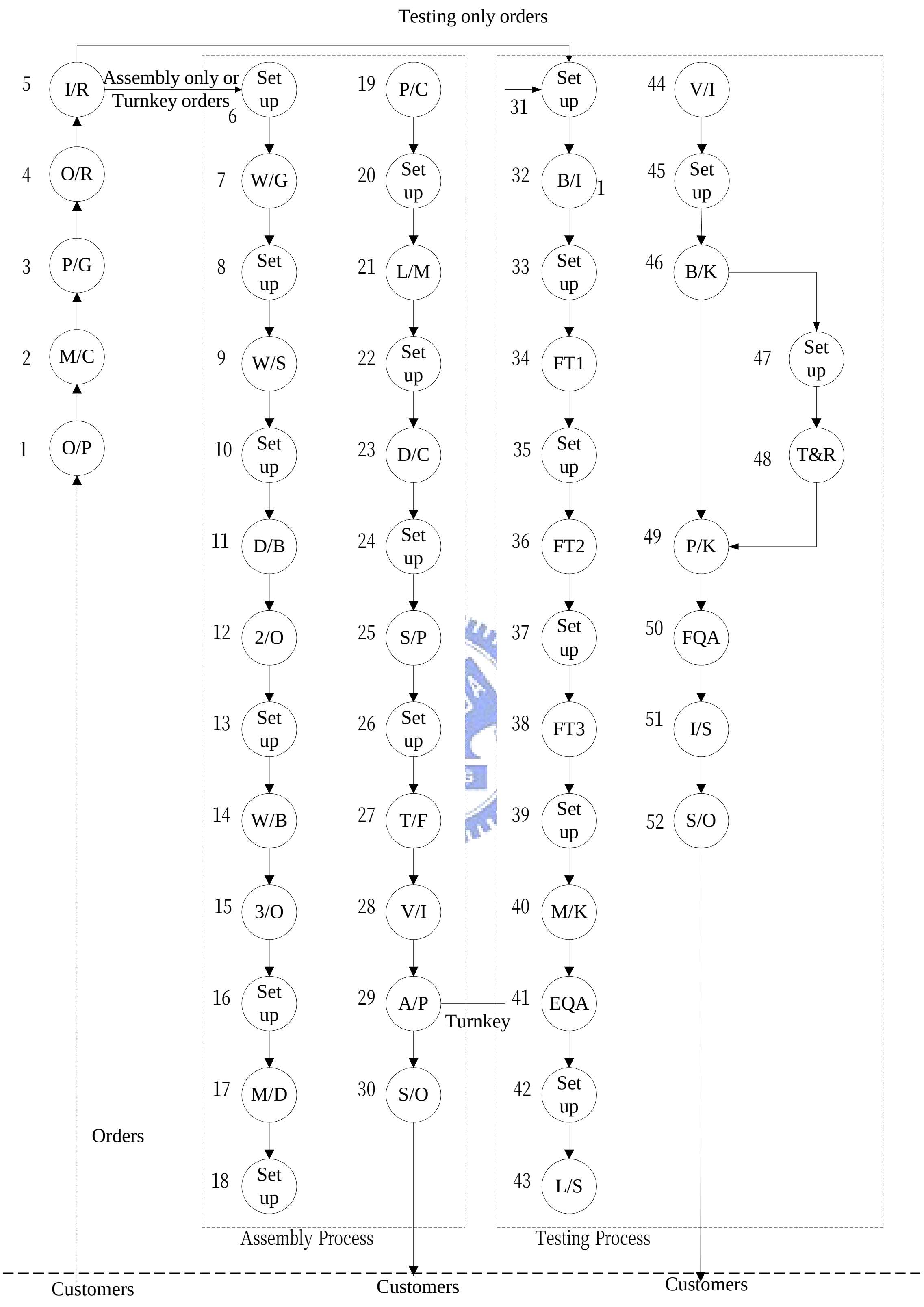


Figure 4.2 Assembly and testing process

4.1.3 Analyze the Activities/Resources from the Process

Based on the manufacturing processes described in the previous section, the activities and resource(s) required and released for each activity are listed in tables 4.1, 4.2 and 4.3, each for indirect manufacturing activities, assembly and testing firm respectively. An activity code is assigned for each activity.

Table 4.1 Define and analyze the indirect manufacturing activities

Activity Code (a)	Activity	Resource(s) required F(a)	Resource(s) released F'(a)
1	O/P	Production planner	Production planner
2	M/C	Production planner	Production planner
3	PT/C	Production planner	Production planner
4	O/R	Production planner	Production planner
5	I/R	Operator	Operator
	Facility level activities		

Table 4.2 Define and analyze the activities/resources for assembly firm

Activity Code (a)	Activity	Resource(s) required F(a)	Resource(s) released F'(a)
7	W/G	Grinder, Tape, Wheel	Grinder
9	W/S	Sawing machine, Dicing tap	W/S cost
11	D/B	Die bonder, Lead frame Magazine	Die bonder
12	2nd Optical	Inspector	Inspector
14	W/B	Wire bonder, Gold wire Heat block	Wire Bonder
15	3rd Optical	Inspector	Inspector
17	M/D	Molder, Compound, Carrier	Molder, Magazine
19	P/C	Oven	Oven
21	L/M	Laser Marker	Laser Marker
23	D/C	Dambar-Cutter	Dambar-Cutter
25	S/P	Solder Plating, Solder ball	S/P machine

27	T/F	Trim/Form machine, Tray/Tube	Trim/Form machine, Carrier
28	VI	Inspector	Inspector
29	P/K	Operator, Packing box	Operator
30	S/O	Ship out	Operator
Sa	Set-up	Operator Machine	Operator Machine

Sa={6, 8, 10, 13, 16, 18, 20, 22, 24, 26}

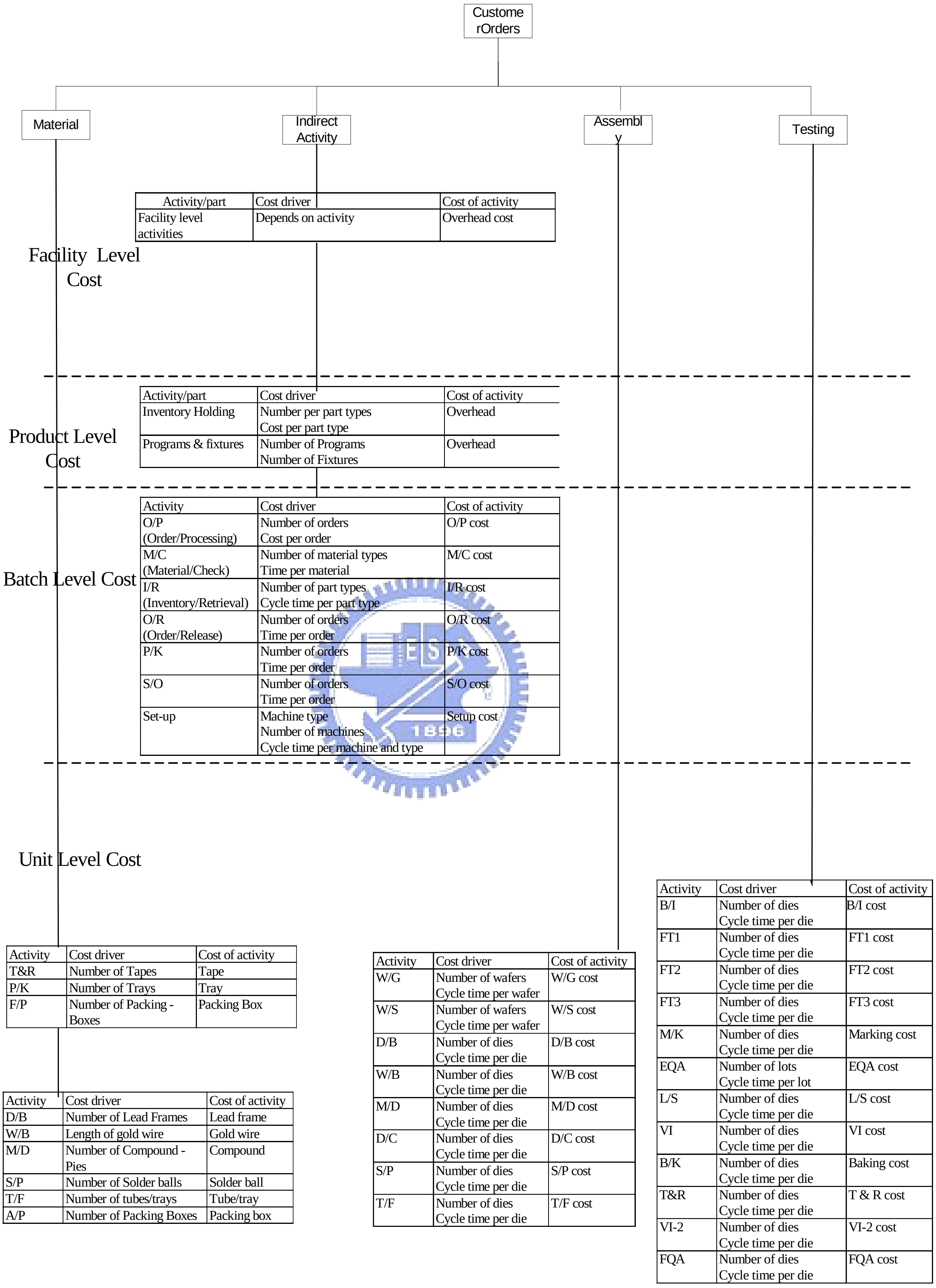
Table 4.3 Define and analyze the activities/resources for testing firm

Activity Code (a)	Activity	Resource(s) required F(a)	Resource(s) released F'(a)
32	B/I	Burn-In board, Program	Burn-in board, Program
34	FT1	Tester, Hi-fix, Program, Gas	Tester, Hi-fix, Program
36	FT2	Tester, Hi-fix, Program	Tester, Hi-fix, Program
38	FT3	Tester, Hi-fix, Program	Tester, Hi-fix, Program
40	M/K	Laser marker	Laser marker
41	EQA	Tester, Inspector	Tester, Inspector
43	L/S	Lead scanner	Lead scanner
44	VI	Inspector	Inspector
46	B/K	Oven	Oven
48	T&R	T&R machine	T&R machine
49	P/K	Operator	Operator
50	FQA	Inspector	Inspector
51	I/S	Storage space	Storage space
52	S/O	Operator	Operator
St	Set-up	Operator, Machine	Operator, Machine

St={31, 33, 35, 37, 39, 42, 45, 47}

4.1.4 Analyze Cost Driver and Cost Object

The cost object used in this case is CO, Figure 4.3 displays the analyzed results of cost drivers and activity costs for semiconductor backend turnkey operations in a collaborative environment. Each activity is likely to have a different cost driver that represents the consumption of the resources. Different cost drivers will create different ways of the calculation of the costs. But it should be noted, values of activity costs vary with type of resource actual consumed even though those activities have the same cost driver. Take the W/B activity as an example, the process cycle time per die depends on what type of wire-bonder machines are actually used; therefore, the cost value result is different.



Activity/part	Cost driver	Cost of activity
Facility level activities	Depends on activity	Overhead cost

Activity/part	Cost driver	Cost of activity
Inventory Holding	Number per part types Cost per part type	Overhead
Programs & fixtures	Number of Programs Number of Fixtures	Overhead

Activity	Cost driver	Cost of activity
O/P (Order/Processing)	Number of orders Cost per order	O/P cost
M/C (Material/Check)	Number of material types Time per material	M/C cost
I/R (Inventory/Retrieval)	Number of part types Cycle time per part type	I/R cost
O/R (Order/Release)	Number of orders Time per order	O/R cost
P/K	Number of orders Time per order	P/K cost
S/O	Number of orders Time per order	S/O cost
Set-up	Machine type Number of machines Cycle time per machine and type	Setup cost

Activity	Cost driver	Cost of activity
T&R	Number of Tapes	Tape
P/K	Number of Trays	Tray
F/P	Number of Packing - Boxes	Packing Box

Activity	Cost driver	Cost of activity
D/B	Number of Lead Frames	Lead frame
W/B	Length of gold wire	Gold wire
M/D	Number of Compound - Pies	Compound
S/P	Number of Solder balls	Solder ball
T/F	Number of tubes/trays	Tube/tray
A/P	Number of Packing Boxes	Packing box

Activity	Cost driver	Cost of activity
W/G	Number of wafers Cycle time per wafer	W/G cost
W/S	Number of wafers Cycle time per wafer	W/S cost
D/B	Number of dies Cycle time per die	D/B cost
W/B	Number of dies Cycle time per die	W/B cost
M/D	Number of dies Cycle time per die	M/D cost
D/C	Number of dies Cycle time per die	D/C cost
S/P	Number of dies Cycle time per die	S/P cost
T/F	Number of dies Cycle time per die	T/F cost

Activity	Cost driver	Cost of activity
B/I	Number of dies Cycle time per die	B/I cost
FT1	Number of dies Cycle time per die	FT1 cost
FT2	Number of dies Cycle time per die	FT2 cost
FT3	Number of dies Cycle time per die	FT3 cost
M/K	Number of dies Cycle time per die	Marking cost
EQA	Number of lots Cycle time per lot	EQA cost
L/S	Number of dies Cycle time per die	L/S cost
VI	Number of dies Cycle time per die	VI cost
B/K	Number of dies Cycle time per die	Baking cost
T&R	Number of dies Cycle time per die	T & R cost
VI-2	Number of dies Cycle time per die	VI-2 cost
FQA	Number of dies Cycle time per die	FQA cost

Figure 4.3 Cost drivers and activity costs for assembly and testing firm

4.2 Develop the ABC Pr/Tr Net

When the ABC data analysis is completed, the simulation model is developed to simulate the manufacturing process and roll up the cost for each order.

4.2.1 Analyze the Activity Transition

Some specific characteristics of the testing process, which impact the normal process are described as follows:

- If high defect results come out from FT1 operation, then rework it.
- After dry the dies (B/K operation), if package type for this lot is tape and reel, then the dies will move to tape and reel operation (T&R), otherwise lot will forward to packing operation (P/K) directly.

According to the above analysis, an activity transition diagram illustrated in Figure 4.4 is used to describe the possible transition state for each activity.

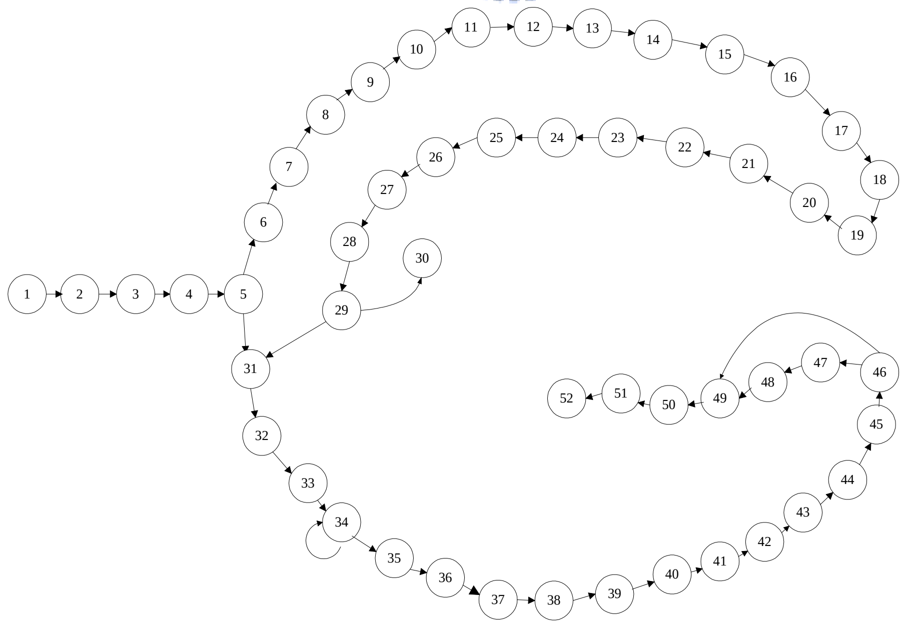


Figure 4.4 Activity transition diagram

4.2.2 Group the Activities into Activity Sets

Refer to tables 4.1, 4.2 and 4.3, which describe the resource(s) required and released for each activity, Functor F(a) is used to supply the resource(s) and F`(a) is used to return resource(s). Moreover, it shows that all activities need resource(s) for operation and after an activity is done, resource(s) must be returned. Split lot happened when activities 4 (Order release), 11 (Die bond) and 37 (FT3) finished, and merge lots occurred after activities 29 and 48 (Packing), therefore special treatments are necessary for lots after these activities (three kinds of activity sets for Z needed to be defined). According to the methodology we defined in section 3.2, three sets; X, Y, Z can be derived:

X: X={1,2…,,52}; the universal set of the activities

Y: Y={Y_i ; $i=1,m$}, in our application, we only concern about whether activity needs resource(s) supply, and since all activities need resource supply, therefore, one set is sufficient for modeling, we set m=1, hence Y={Y_1}. And, since the exclusive relationship$Y_1 \cup Y_2 \cup \ldots \cup Y_m$=X hold, therefore Y_1 = X = {1,2…,52}.

Z: Z={Z_j ; $j=1,n$}, in our application, we concern on three different changes of lots after some activities, i.e. the split, merge and neither split nor merge. Therefore, activities are categorized into three subsets, given $j=3$, hence Z={Z_1, Z_2, Z_3}. Each subset in Z is defined as follows:

Z_1={4,11,37}; the split set

Z_2={29,48}; the merge set

and since the exclusive relationship $Z_1 \cup Z_2 \cup Z_3$ = X hold, therefore

Z_3=X- Z_1 - Z_2= {1,2…,,52}-{4,11,37}-{29,48}; the neither split nor merge set

The X, Y and Z set will be used to construct the simulation model in the next section.

4.2.3 Develop the ABC Pr/Tr Net Simulation Model

By applying the core components defined below, as depicted in Figure 4.5, an ABC Pr/Tr Net was constructed to simulate a dynamic resources consuming process for semiconductor backend operations.

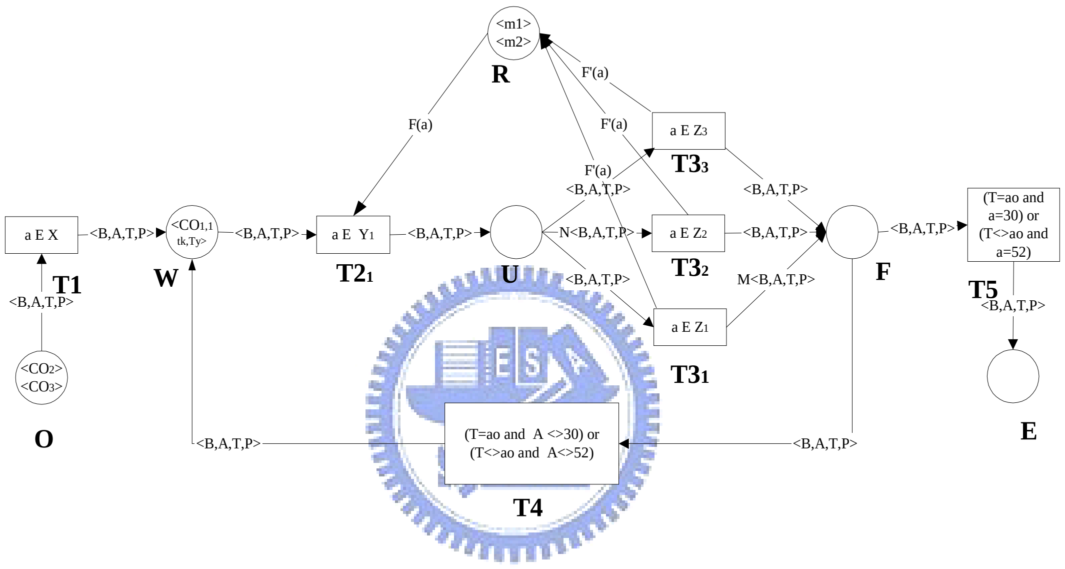


Figure 4.5 The ABC Pr/Tr Net model

- Predicate: P={R, O, W, U, F, E}
- Transition: T={T1, T2$_{1}$, T3$_{j=1,3}$, T4, T5, T6} are transitions with a logical formula.
- Label: the labels used in the case is <B, A, T, P> for LB1, LB2$_{1}$, LB3$_{j=1,3}$, LF4 and LB5. Four attributes are used in this label, which are defined as below:

  B: Batch number

  A: Activity code; values for A={1,2… ,52}

  T: Product type; values for T={ao, to, tk}

  ao: assembly only

to: testing only

tk: turnkey

P: Package type; values for P={Ty, T&R}

Ty : Tray

T&R: Tape and Reel

- Logical formula: in this case, logical formulas are defined as:

LF1 : $a \in X$; is the logical formula used to describe the logic condition to release an order into production.

$LF2_1$: $a \in Y_1$; is the logical formula used to control the trigger of the transition $T2_1$ and transit a lot to U predicate for resource(s) consuming.

$LF3_1$: $a \in Z_1$; is the logical formula used to control the trigger of transition $T3_1$ which split and transit lots from U to F predicate. (for split activity set)

$LF3_2$: $a \in Z_2$; is the logical formula used to control the trigger of transition $T3_2$ which merge and transit lot from U to F predicate. (for merge activity set)

$LF3_3$: $a \in Z_3$; is the logical formula used to control the trigger of transition $T3_3$ and transit the lot from U to F predicate. (for neither split nor merge set)

LF4: (T='ao' and A < 30) or (T<>'ao' and A<52); is a logical formula to judge the condition for lots transit to W predicate for the next activity (transit from F to W predicate).

LF5: (T='ao' and A = 30) or (T<>'ao' and A=52); is a logical formula to judge the condition for lots to exit the system (transit from F to E).

4.2.4 Profit Calculation

The manufacturing cost considered in this case includes direct labor cost, direct material cost, and some overhead costs derived directly by the cost drivers, which were defined in section 3.2. The Manufacturing Net Profit may express as follows:

$$\text{MNP}=\text{TSR}-\text{TMC}=(\text{ATR}+\text{TTR}+\text{KTR}) - [(\text{ATC}+\text{TTC}+\text{KTC})-\text{DM}]$$

$$=\left(\sum_{i=1}^{m} A_i Q_i + \sum_{i=1}^{p} B_i Q_i + \sum_{i=1}^{s} C_i Q_i \right) - \left[\left(\sum_{i=1}^{m}\sum_{j=1}^{n}\sum_{k=1}^{o} Q_{ij} C_{ijk} + \sum_{i=1}^{p}\sum_{j=1}^{q}\sum_{k=1}^{r} Q_{ij} C_{ijk} + \sum_{i=1}^{s}\sum_{j=1}^{t}\sum_{k=1}^{u} Q_{ij} C_{ijk} \right) - \right.$$

$$\left.\left(\sum_{i=1}^{m}\sum_{j=1}^{n}\sum_{k=1}^{o} Q_{ij} M_{ijk} + \sum_{i=1}^{p}\sum_{j=1}^{q}\sum_{k=1}^{r} Q_{ij} M_{ijk} + \sum_{i=1}^{s}\sum_{j=1}^{t}\sum_{k=1}^{u} Q_{ij} M_{ijk} \right)\right]$$

Variables defines bellow:

MNP: Manufacturing Net Profit

TSR : Total Sales Revenue

TMC: Total Manufacturing Cost

ATR : Assembly Total Revenue

TTR : Testing Total Revenue

KTR : Turnkey Total Revenue

ATC : Assembly Total Cost

TTC : Testing Total Cost

KTC : Turnkey Total Cost

DM : Direct Material Cost

i: *i* th Customer Order (CO)

j: *j* th Manufacturing Order (MO)

k: *k* th activity

m: number of customer orders for assembly only

n: number of manufacturing orders per customer order for assembly only

o: number of activities per lot for assembly only

p: number of customer orders for testing only

q: number of manufacturing orders per customer order for testing only

r: number of activities per lots for turnkey

s: number of customer orders for turnkey

t: number of manufacturing orders per customer order for turnkey

u: number of activities per lots for turnkey

A_i: sales price of customer order i per piece for assembly only

B_i: sales price of customer order i per piece for testing only

C_i: sales price of customer order i per piece for turnkey

Q_i: quantity of customer order i

Q_{ij} : quantity of manufacturing order j for customer order i

C_{ijk}: cost value of cost driver for activity k, manufacturing order j and customer order i. C_{ijk} is determined by resource available dynamically in shop floor.

M_{ijk}: direct material cost of activity k for manufacturing order j and customer order i.

The relative MNP of the set of release rules and dispatching rules used in shop floor for the mix of released COs can be rolled up and estimated by running the ABC/CPPS and using the formula defined above. And it should be noted that C_{ijk} is dynamically determined.

CHAPTER 5

IMPLEMENTATION OF ABC/CPPS

5.1 Design of the Data Structure

The data structure used in this study is an object-oriented relationship concept, four kinds of data structures are important to design the system, the resource token, the order token, the rule and the schedule data structure. As shown by Figure 5.1(a), a data structure for **resource** token is organized by a predicate resource, and other data items related to resource, such as the company_ame, the plant_name, the resource_type, the resource_id, the time_available_from, the time_available_to and the cost are all included. The company_name and the plant_name are used to denote the name of a company and a plant respectively, and the resource_type is used to denote the information of a specific machine type (such as wafer_grinder, wafer_saw, wire_bonder, etc.) that is commonly used in semiconductor backend shop floor. Resource_id is a data item used to keep the identification number of a specific resource, and the time_available_from data item specifies the beginning available time of this resource while data item time_available_to specifies the end of available of time of this specific resource. A resource token data structure in the data base is represented as resource(company_name, plant_name, resource_type, resource_id, time_avail_for m, time_avail_to).

The data structure for an **order** token is designed as shown by Figure 5.1(b) and represented as order(customer, order_no, product_type, sale_price, quantity, due_date, assign_company, assign_ plant) in the database, the customer data item is used to identify the customer, order_no is used to keep the information of order

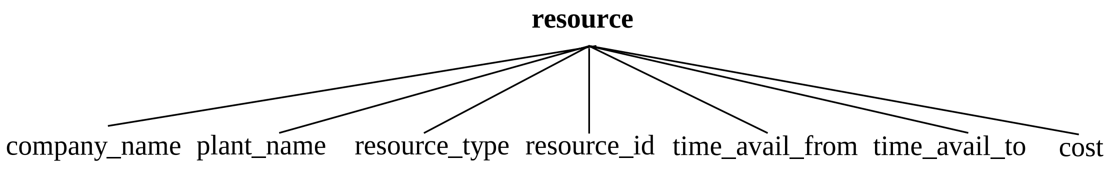


Figure 5.1(a) Data structure for a resource token

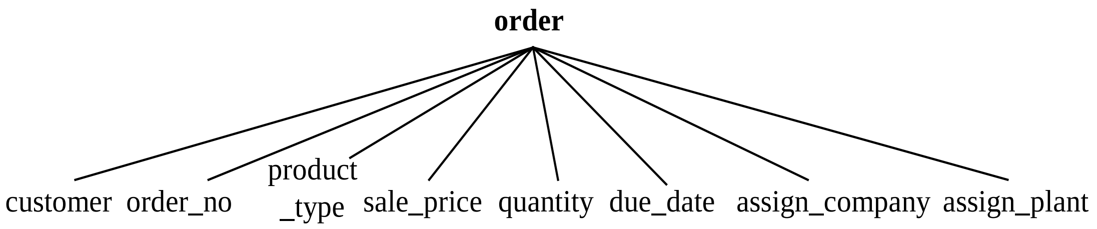


Figure 5.1(b) Data structure for an order token

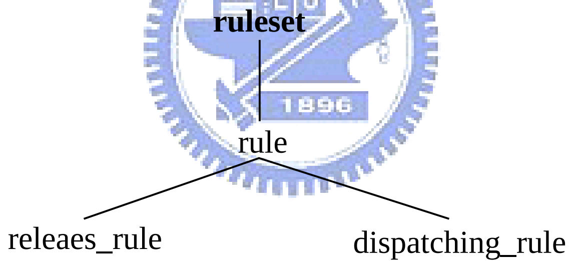


Figure 5.1(c) Data structure for rule

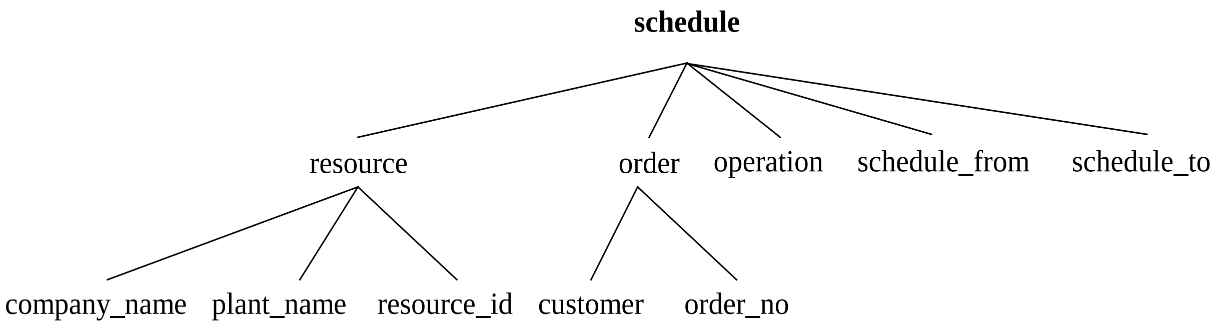


Figure 5.1(d) Data structure for schedule

number, the assign_company is used to keep the information of the company previously assigned, and the assign_plant is used to keep the information of the plant of a specific company previously assigned.

The data structure for **ruleset** is designed as shown by Figure 5.1(c) and represented in database as ruleset(release_rule, dispatching_rule), the release_rule data item is used to keep the information of release rule used while the dispatching_rule is used to keep the dispatching rule during the simulation.

Finally, one more important data structure needs to be defined is the **schedule** data structure. This data structure keeps the scheduling results from the assignment. As shown by Figure 5.1(d), the data structure for schedule is represented as schedule(resource(company_name, plant_name, resource_id), order(customer, order_no), operation, schedule_from, schedule_to).

5.2 Implementation of the ABC Pr/Tr Net

The second step to implement the ABC/CPPS system is to extract the knowledge (facts and rules) from the designed ABC Pr/Tr net as shown in Figure 4.5. Basically, the knowledge base is composed by predicate knowledge (facts) and transition knowledge (rules), thus six steps for bottom-up implementation of the system are proposed: (1) extract predicate knowledge, (2) extract the static transition logic rules, (3) transform the static transition logic rules to dynamic transition logic rules, (4) enable the iterative firings of dynamic transition logic rules, (5) incorporate other expert knowledge to enhance system capability and (6) design goals for this expert system to explore. Detail contents are described below.

1. **Extract predicate knowledge**: According to the Figure 4.5, there are five predicates (O, E, W are used to describe the possible states for an order, and R

is used for resource only, U is used for both order and resource) in the ABC Pr/Tr Net. Take the O predicate as an example, for predicate O to be true, there must exist a token O(B,A,T,P), in which token variables B,A,T,P are bound to token values. (Token is represented by a fact in database). If expressed by first order predicate logic, this rule can be written as

O(B,A,T,P):-O_token(B,A,T,P).

Applying similar explanations to other predicates, then some other predicate logic rules can be derived.

```
U(B,A,T,P):-U_token(B,A,T,P).
      R(A):-R_token(B,A,T,P).
F(B,A,T,P):-F_token(B,A,T,P).
E(B,A,T,P):-E_token(B,A,T,P).
W(B,A,T,P):-W_token(B,A,T,P).
```

2. **Extract the static transition rules**: Static transition rules can be visually derived directly from Pr/Tr net. According to the ABC Pr/Tr Net, there are seven transitions, the T1, $T2_1$, $T3_1$, $T3_2$, $T3_3$, T4 and T5 transition in the net. Take the T1 transition as an example, the transition is firabe only if predicate O is true. The expression: O(B,A,T,P) → W(B,A,T,P) implies W(B,A,T,P) is true only if O(B,A,T,P) is true, furthermore, for O predicate to be true, there must exist a O_token(B,A,T,P) which is defined in step 1. If expressed by logic syntax with Visual Prolog, this logic is represented as rule 1 in Table 5.1. Similarly, other static transition logic rules can be derived and tabulated.
3. **Transform the static transition logic rules to dynamic transition logic rules**: Each transition rule in table 5.1 is modified by appending two predicates (the retract and the assert predicate) to transform the static transition rules into dynamic transition rules. Retract predicate is used to disappear the fact (token) in the dynamic database, and assert predicate is used

to appear another fact into the database. This simulates the transition behaviors of the task and resources tokens in the Pr/Tr net. Note that an additional predicate get_next_activity(A, Next_A) is designed and used to retrieve the next activity in rule 2, the variable A is a input argument indicates the current activity while Next_A is the output variable to indicate the next activity to be processed. Also, In rule 2, for transition T2 to be firable, avail_R(A) (is used to retrieve the available resource(s) for activity A) must be satisfied, too. Moreover, an equation formula, A=LA, is used to compare the current activity with the constant variable LA that indicates the last activity number. If last activity is reached, then transition T5 is triggered and puts the order token to E predicate. The whole dynamic transition logic rules are tabulated in table 5.2.

Table 5.1 The static transition logic rules for ABC Pr/Tr Net model

Transition	Rule No.	Rule contents (predicate knowledge)
T1	Rule 1	W(B,A,T,P):-O(B,A,T,P), $A \in X$.
T2	Rule 2	U(B,A,T,P):-W(B,A,T,P), $A \in Y_{1.}$ avail_R(A) .
$T3_1$	Rule 3	F(B,A,T,P):-U(B,A,T,P), $A \in Z_{1.}$
$T3_2$	Rule 4	F(B,A,T,P):-U(B,A,T,P), $A \in Z_{2.}$
$T3_3$	Rule 5	F(B,A,T,P):-U(B,A,T,P), $A \in Z_{3.}$
T4	Rule 6	W(B,A,T,P):-F(B,A,T,P), T=”ao”, A<>30. W(B,A,T,P):-F(B,A,T,P), T<>”ao”, A<>52.
T5	Rule 7	E(B,A,T,P):-F(B,A,T,P), T=”ao”, A=30. E(B,A,T,P):-F(B,A,T,P), T<>”ao”, A=52.

Table 5.2 The dynamic transition logic rules for ABC Pr/Tr Net model

Transition	Rule No.	Rule contents (predicate knowledge)
T1	Rule 1	W(B,A,T,P):-O(B,A,T,P), $A \in X$. retract(O_token(B,A,T,P)), asserta(W_token(L,Next_A,R,T)).
T2	Rule 2	U(B,A,T,P):-W(B,A,T,P), $A \in Y_1$, avail_R(A), get_next_activity(A,Next_A), retract(W_token(B,A,T,P)), asserta(U_token(B,A,T,P)).
$T3_1$	Rule 3	F(B,A,T,P):-U(B,A,T,P), $A \in Z_1$, retract(U_token(B,A,T,P)), asserta(F_token(B,A,T,P)), asserta(F'(A)).
$T3_2$	Rule 4	F(B,A,T,P):-U(B,A,T,P), $A \in Z_2$, retract(U_token(B,A,T,P)), asserta(F_token(B,A,T,P)), asserta(F'(A)).
$T3_3$	Rule 5	F(B,A,T,P):-U(B,A,T,P), $A \in Z_3$, retract(U_token(B,A,T,P)), asserta(F_token(B,A,T,P)), asserta(F'(A)).
T4	Rule 6	W(B,A,T,P):-F(B,A,T,P), T="ao", A<>30, retract(F_token(B,A,T,P)), asserta(W_token(B,A,T,P)). W(B,A,T,P):-F(B,A,T,P), T<>"ao", A<>52, retract(F_token(B,A,T,P)), asserta(W_token(B,A,T,P)).
T5	Rule 7	E(B,A,T,P):-F(B,A,T,P), T="ao", A=30, retract(F_token(B,A,T,P)), asserta(E_token(B,A,T,P)). E(B,A,T,P):-F(B,A,T,P), T<>"ao", A=52, retract(F_token(O,A,R,T)), asserta(E_token(B,A,T,P)).

4. **Enable the iterative firings of dynamic transition logic rules**: In order for the four dynamic transition rules to be fired iteratively until all tasks are completely assigned, a mechanism is designed. A super transition transition(R,

P), which replacing the heads (predicate on the left side of symbol :-) of the four clause rules, is used to invoke the pre-conditions (predicates) of T1, $T2_1$, $T3_1$, $T3_2$, $T3_3$, T4 and T5 iteratively until the boundary condition, (no O_token) ^ (no W_token) ^ (no U_token) ^ (no F_token) , is true. This implies all order tokens are at closed_task predicate, i.e. all orders are planned therefore the scheduling work may stop. With this desired mechanism, the iterative firings of transitions could be simulated and modeled. Note, variable R used in transition(R, P) specifies the dispatching rule used, and variable P denotes the outcome profit.

```
transition(R,P):- not(O_token(_)),not(W_token(_)),not(U_token(_)),
                  not(F_token(_)),!.
transition(R,P):- O(B,A,T,P), A∈X.
                   retract(O_token(B,A,T,P)),
                   asserta(W_token(L,Next_A,R,T)).
transition(R,P):-.W(B,A,T,P), A∈Y1,
                   get_next_activity(A,Next_A),
                   avail_R(A),
                   retract(W_token(B,A,T,P)),
                   asserta(U_token(B,A,T,P)).
transition(R,P):- U(B,A,T,P), A∈Z1,
                   retract(U_token(B,A,T,P)),
                   asserta(F_token(B,A,T,P)),
                   asserta(F'((A)).
transition(R,P):- U(B,A,T,P), A∈Z2,
                   retract(U_token(B,A,T,P)),
                   asserta(F_token(B,A,T,P)),
                   asserta(F'((A)).
transition(R,P):- U(B,A,T,P), A∈Z3,
                   retract(U_token(B,A,T,P)),
                   asserta(F_token(B,A,T,P)),
                   asserta(F'((A)).
transition(R,P):- F(B,A,T,P),
                   A< LastActivity,
                   retract(F_token(B,A,T,P)),
                   asserta(W_token(B,A,T,P)),
transition(R,P):- F(B,A,T,P),
                   A=LastActivity,
                   retract(F_token(B,A,T,P)),
                   asserta(E_token(B,A,T,P)),
transition(R,P):-transition(R,P).
```

5. **Incorporate other expert knowledge to enhance system capability**: Expert knowledge, which can enhance or make system more intelligent, can be incorporated in this step, such as knowledge to retrieve the next activity if expressed by logic rule can be represented as

```
get_next_activity(Activity,Next_activity):- Next_activity=Activity+1.
```

6. **Design the goals for this expert system to explore**: The final step (highest level work) is to design the goals for this system to explore. During the reasoning process of the goal, transition rules are invoked by simulate(Rule, Profit) in order to simulate the task assignment process. When a schedule is planned and the outcomes compliant to the goal, then schedule is feasible, otherwise suggestions may be provided by the knowledge base. Two goal examples are given below to give more detail explanation.

```
Goal 1: simulate_one(Rule,Profit):-ruleset(Rule), simulate(Rule,Profit).

Goal 2: simulate_all(Rule,Profit):-ruleset(Rule), simulate(Rule,Profit),
                                    max_ profit (Max_Rule,Max_Profit).

simulate(Rule,Profit):-transition(Rule,Profit), fail.
simulate(Rule,Profit).
```

A reasoning tree is depicted by Figure 5.2(a), which gives an example for the expert system to explore the goal 1. Goal 1 is simply designed to implement the single use of a dispatching rule. For the clause that refers to goal 1 to be true, the condition, predicate ruleset(Rule) must be satisfied firstly, therefore ruleset(Rule) unifies into the database by the depth first search mechanism embedded in logic program, and then variable Rule binds to the value ”rule1”, after passing this value to the 2nd predicate simulate(Rule, Profit) in the Hone clause, simulate(“rule1”, Profit) is then tried to be satisfied secondly, this incurs the iterative firings of the transitions in the ABC Pr/Tr Net until the boundary clause (condition) is satisfied. Then the output variable, Profit, indicates the profit information by using the dispatching rule, “rule1”. According to ABC theory, analyzing the cost drivers can derive the cost for each activity. Therefore, if the sales price for each order is known, then the profit can be calculated.

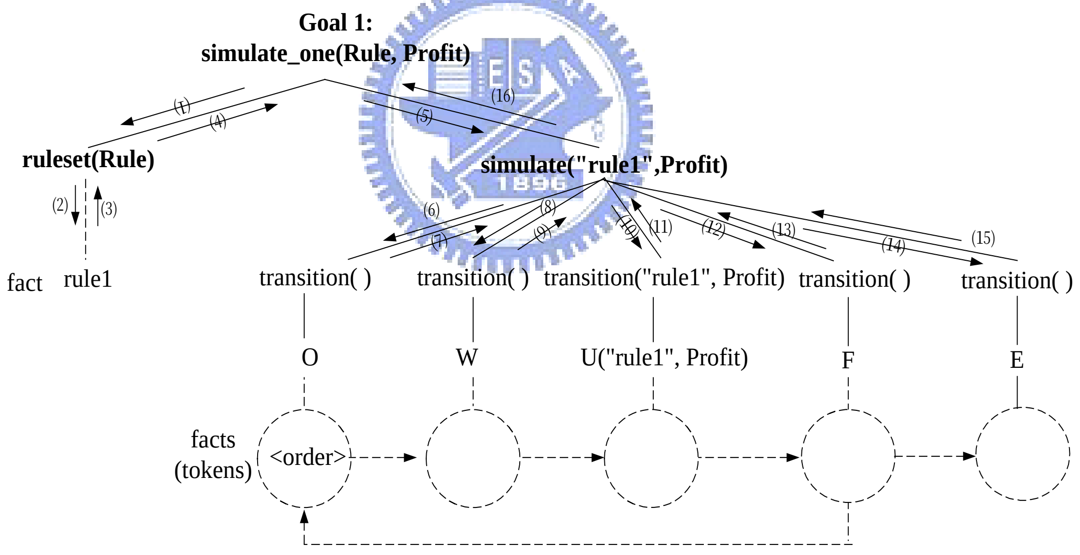


Figure 5.2(a) A reasoning tree for goal 1

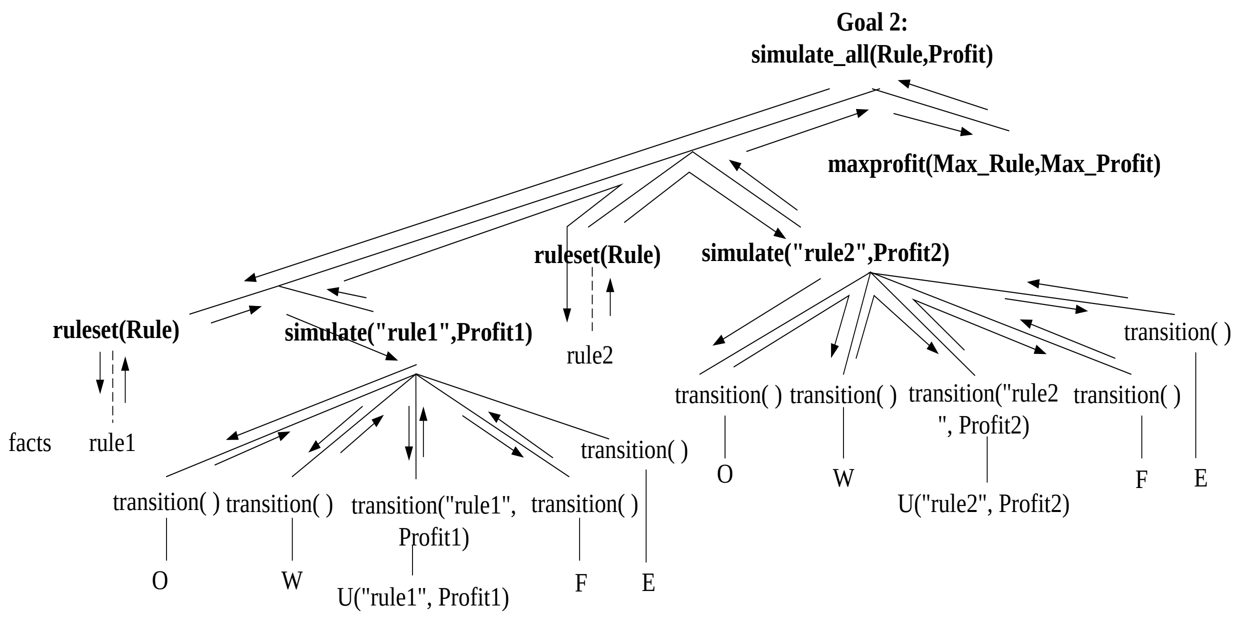


Figure 5.2(b) A reasoning tree for goal 2

To extend the exploration of alternative solutions, goal 2 is designed to illustrate the search of the solution space for finding another superior solution in the point view of maximum profit. Figure 5.2(b) shows the reasoning sequences in a tree for all the available dispatching rules. Firstly, “rule1” is unified by predicate ruleset(Rule) and bound to variable Rule, and then simulate(“rule1”, Profit1) is invoked and triggering the fires of transitions. A collaborative schedule will be created and profit information (Profit 1) will come out after a simulation run. In the next step, ruleset(Rule) unifies into database and binds the variable Rule with value “rule2”, simulate(“rule2”,Profit2) is invoked again, and profit information (Profit2) is got. Lastly, the final predicate maxprofit(Max_Rule, Max_Profit) is executed to suggest the superior solution from the alternatives and the collaborative schedule is suggested, too. Obviously, rules are appendable, and other goals can be easily set up to extend or empower the system capability.

5.3 System Structure of ABC/CPPS

As depicted in Figure 5.3, there are three main components in this expert scheduling system, the Graphic User Interface (GUI), the collaborative scheduling model (Model) and the knowledge database (KB) module. The collaborative scheduling module with simulation knowledge and expert knowledge incorporated is the kernel of this system. The GUI that is developed by Visual Prolog, providing a window environment to select a goal (or query) for reasoning, is shown in Figure 5.4. Collaborative schedule is created after the simulation run and passed to a web site for directly accessing from the Internet, or data in XML format could be created for data exchange among the collaborative members.

In practice, the expert scheduling system is useful both in internal and/or collaborative scheduling. For internal use, each collaborative member can run its own autonomous schedule first, and then release the free capacity (collaborative resources) to virtual enterprise for further creating the collaborative manufacturing schedule. The two-stage procedure keeps both the autonomous and collaborative planning feasible.

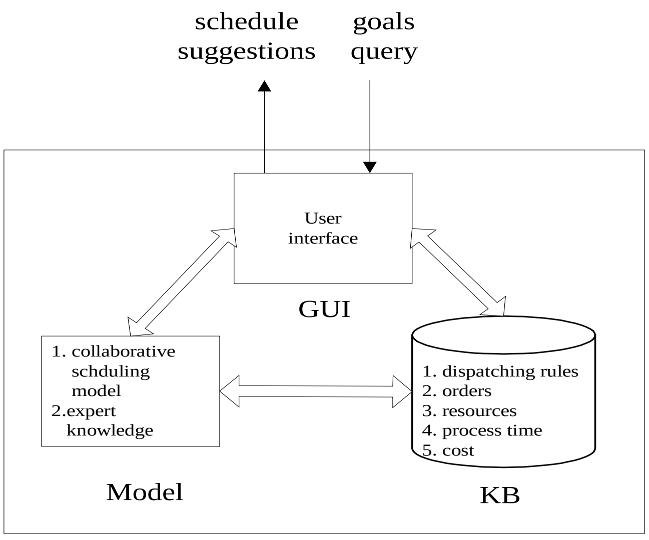
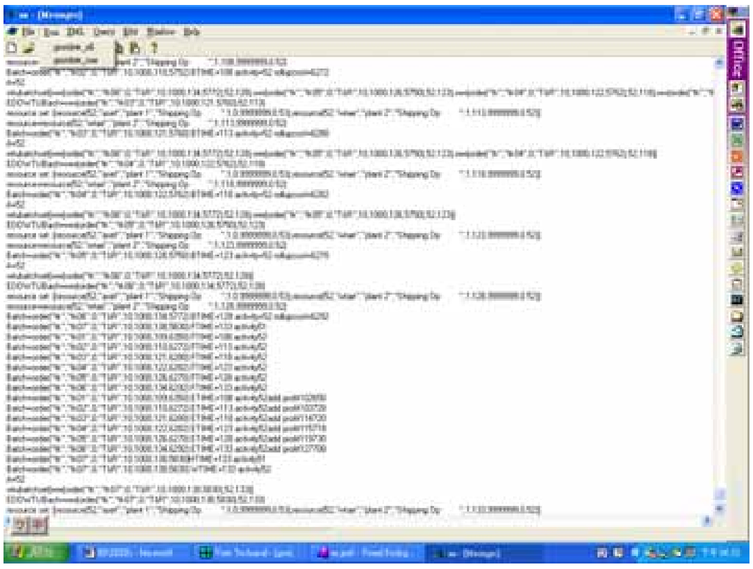


Figure 5.3 The system structure for ABC/CPPS



[illegible]

Figure 5.4 The GUI of the ABC/CPPS

5.4 Web Publish and Data Exchange

To facilitate the implementation of collaborative production, a collaborative production planning system (CPPS) is required. As shown by Figure 5.5, a collaborative planning model is built to serve as the planning engine based on available collaborative resources. After the collaborative manufacturing schedule is planned, the results may be either published to web server which is directly assessable by browser from the client, or transferred to each member by using XML-formatted file. Based on the web-server architecture and the Internet infrastructure, collaborative members can be networked through the Internet to provide a collaborative scheduling/manufacturing environment.

About the markup language, Standard Generalized Markup Language (SGML), HyperText Markup Language (HTML) and eXtensible Markup Language (XML) are the three most important ones. Each language has a unique purpose. SGML is a rich meta language that is useful for defining an almost endless supply of markup languages. HTML is useful for displaying Web pages. XML, is the extensible markup language, and was developed from SGML. Now

XML is becoming a standard way for people to export and import data from different systems. Its simplicity makes it easy for different systems to exchange data. Details about the XML data transformation in ABC/CPPS are presented in next chapter.

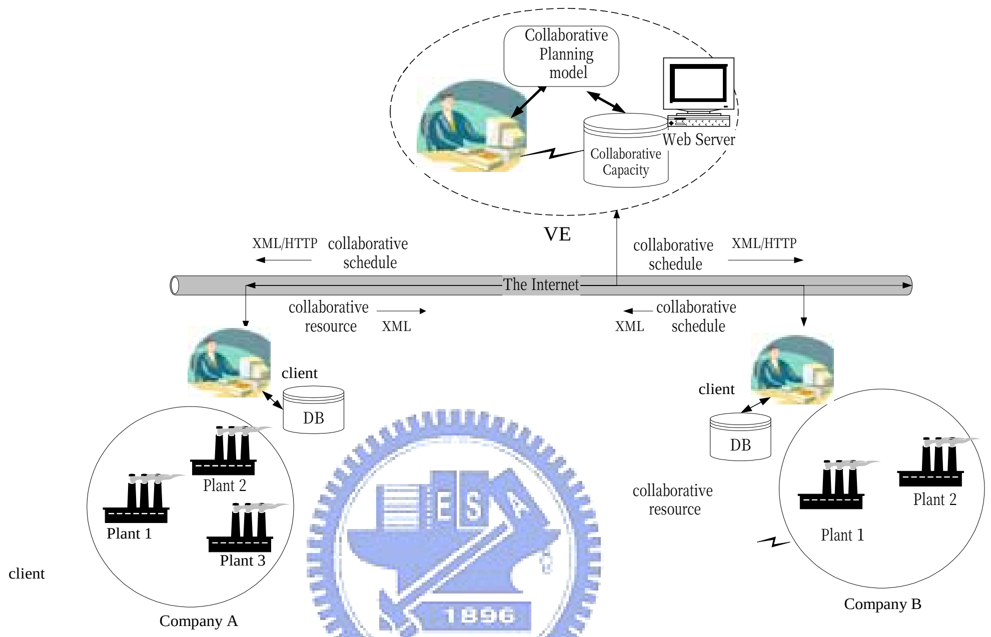


Figure 5.5 Client/Server wet site structure

CHAPTER 6

NUMERICAL EXAMPLE

6.1 The Scenario

As depicted by Figure 6.1, in this scenario example three firms are included; an assembly firm (ASE) and two testing firms (the ASET and the WTAE). The three firms are allied together as a VE to fulfill the customers' orders. Available collaborative resource(s) are declared by each firm, and then collaborative schedules are going to be planned out for all the partners to cooperate.

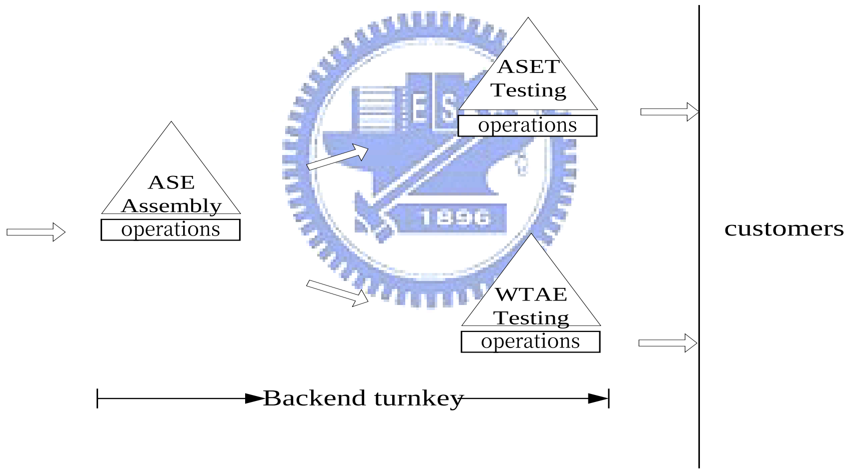


Figure 6.1 The scenario

6.2 The Input Data

According to the process described in section 4.1.3, Table 4.1(a), 4.2(b) and 4.3(c) show the activities (operations) conduced in the semiconductor backend, and data about the resource required, process time, cost and collaborative operation information are also available for company ASE, ASET and WTAE respectively. Other data following the data structure defined in section 5.1 are also provided.

- **Dispatching rules database:** Two dispatching rules are used in this example. According to the data structure which is designed as ruleset(rule(release_rule, dispatching_rule) in section 5.1, the rules are represented as:

```
ruleset(rule("EDD","lwl").
ruleset(rule("EDD","lct").
```

EDD: the Earliest Due-Date Rule for release rule.
lct: the Least Cost Rule for dispatching rule.
(select the resource with least cost)
lwl: the Least Work Load rule for dispatching rule.
(select the resource with least workload)

- **Order database**: Nine orders are given in this example. According to the data structure which is defined as order (customer, order_no, product_type, sale_price, quantity, due_date, assign_company, assign_plant) in section 5.1, the orders are represented as:

```
order("tk","tk01",0,"T&R",10,1000,109,0).
order("tk","tk02",0,"T&R",10,1000,110,0).
order("tk","tk03",0,"T&R",10,1000,121,0).
order("ao","ao01",0,"T&R",10,1000,81,0).
order("tk","tk04",0,"T&R",10,1000,122,0).
order("tk","tk05",0,"T&R",10,1000,126,0).
order("tk","tk06",0,"T&R",10,1000,134,0).
order("tk","tk07",0,"T&R",10,1000,138,0).
```

- **Collaborative resources database:** The basic collaborative resources data of companies ASE, ASET and WTAE are shown in Tables 6.1(a), 6.1(b) and 6.1(c).

Table 6.1(a) Basic data for assembly firm ASE in example case

Activity code (a)	Activity	Resource required R(a)	Process time	Company ASE		
				Plant/No.	M/No.	Cost
1	O/P	Productoion planner	2	Plant 1	No. 1	0.05
2	M/C	Productoion	3	Plant 1	No. 1	0.05
3	PT/C	planner	4	Plant 1	No. 1	0.05
4	O/R	Productoion	2	Plant 1	No. 1	0.05
5	I/R	planner	3	Plant 1	No. 1	0.05
7	W/G	Wafer Grinder	2	Plant 1	No. 1 No. 2 No. 3	0.15 0.15 0.15
9	W/S	Wafer Saw	4	Plant 1	No. 1 No. 2 No. 3	0.20 0.25 0.22
11	D/B	Die Bonder	2	Plant 1	No. 1 No. 2 No. 3 No. 4 No. 5	0.25 0.25 0.25 0.25 0.25
12	2nd Optical	Inspector	3	Plant 1	No. 1	0.25
14	W/B	Wire Bonder	24	Plant 1	No. 1 No. 2 No. 3 No. 4 No. 5	0.15 0.23 0.25 0.25 0.25
15	3rd Optical	3rd inspector	3	Plant 1	No. 1	0.02
17	M/D	Molder	2	Plant 1	No. 1 No. 2 No. 3	0.05 0.05 0.05
19	P/C	Oven	4	Plant 1	No. 1 No. 2	0.05 0.05
21	L/M	Laser Marker	2	Plant 1	No. 1 No. 2	0.15 0.15
23	D/C	Damber cutter	4	Plant 1	No. 1	0.05
25	S/P	Solder plating	3	Plant 1	No. 1 No. 2	0.05 0.05
27	T/F	Trim/Form	2	Plant 1	No. 1	0.05
28	VI	VI inspector	3	Plant 1	No. 1	0.05
29	P/K	Packing operator	4	Plant 1	No. 1	0.05
30	S/O	Ship out operator	2	Plant 1	No. 1	0.05

.

Table 6.1(b) Basic data for testing firm ASET in example case

Activity code (a)	Activity	Resource required R(a)	Process time	Company ASET		
				Plant/No.	M/No.	Cost
32	B/IP	Burn In oven	3	Plant 1	No.1	0.45
34	FT1	Tester	3	Plant 1	No. 1 No. 2 No. 3	0.15 0.15 0.15
36	PT2	Tester	4	Plant 1	No. 1 No. 2 No. 3	0.20 0.20 0.20
38	FT3	Tester	3	Plant 1	No. 1 No. 2 No. 3	0.29 0.29 0.29
40	M/K	Marker	3	Plant 1	No. 1	0.39
41	EQA	Operator	3	Plant 1	No. 1	0.45
43	L/S	Laser scanner	4	Plant 1	No. 1	0.22
44	V/I	V/I inspector	3	Plant 1	No. 1	0.21
46	2nd Optical	Inspector	4	Plant 1	No. 1	0.25
48	T/R	TR machine	3	Plant 1	No. 1 No. 2	0.51 0.51
49	P/K	P/K machine	4	Plant 1	No. 1	0.32
50	FQA	FQA inspector	3	Plant 1	No. 1	0.23
51	I/S	Inventory storage	5	Plant 1	No. 1	0.36
52	S/O	Ship out	5	Plant 1	No. 1	0.53

.

Table 6.1(c) Basic data for testing firm WTAE in example case

Activity code (a)	Activity	Resource required R(a)	Process time	Company WTAE		
				Plant/No.	M/No.	Cost
32	B/IP	Burn In oven	3	Plant 2	No. 1 No. 2	0.5 0.5
34	FT1	Tester	3	Plant 2	No. 1 No. 2 No. 3 No. 4 No. 5	0.22 0.22 0.22 0.22 0.22
36	PT2	Tester	4	Plant 2	No. 1 No. 2 No. 3 No. 4 No. 5	0.22 0.22 0.22 0.22 0.22
38	FT3	Tester	3	Plant 2	No. 1 No. 2 No. 3 No. 4 No. 5	0.28 0.22 0.25 0.23 0.22
40	M/K	Marker	3	Plant 2	No. 1	0.39
41	EQA	Operator	3	Plant 2	No. 1	0.47
43	L/S	Laser scanner	4	Plant 2	No. 1	0.24
44	V/I	V/I inspector	3	Plant 2	No. 1	0.22
46	2nd Optical	Inspector	4	Plant 2	No. 1	0.42
48	T/R	TR machine	3	Plant 2	No. 1 No. 2	0.52 0.52
49	P/K	P/K machine	4	Plant 2	No. 1	0.32
50	FQA	FQA inspector	3	Plant 2	No. 1	0.22
51	I/S	Inventory storage	5	Plant 2	No. 1	0.35
52	S/O	Ship out	5	Plant 2	No. 1	0.52

.

6.3 The Outcomes

After the simulation run by the expert scheduling system with goal 2, Figure 6.2 depicts the results. The profit by using ruleset(rule("EDD", "lwl",”nonsubcontract”)) is 894,919, and the profit for ruleset(rule("EDD", "lct",”nonsubcontract”)) is 895,332. Therefore the rule ruleset(rule("EDD", "lct",”nonsubcontract”)) is suggested and the collaborative schedules created for each collaborative member to execute are illustrated in appendix A, B and C.

ruleset: rule("EDD","lwl","nonsubcontract") profit= 894919   ETIME=132

ruleset: rule("EDD","lct","nonsubcontract") profit= 895332   ETIME=138

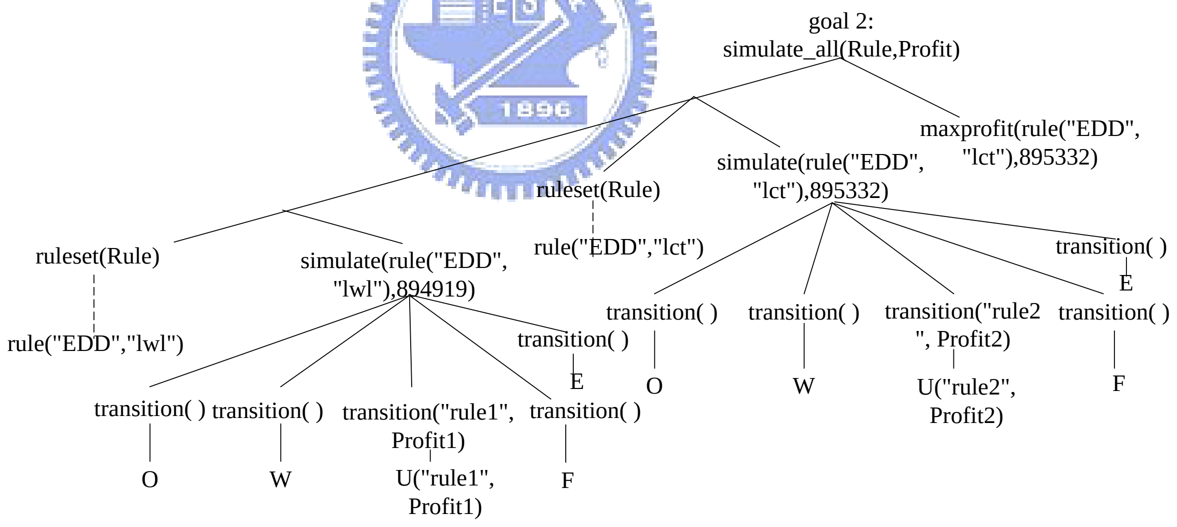


Figure 6.2 The reasoning tree for example case

CHAPTER 7

CONCLUSIONS

7.1 Conclusions

While the market confronts an environment of low profit margin for semiconductor backend, creating a strategic alliance or integrating the enterprise's internal firms by means of collaborative planning/operations mechanism to gain competitive advantage is inevitable. Consequences of non-financial measures used in the past alone are always weak and vague when connected to the enterprise financial objectives; therefore it can be considered to supplement the non-financial measures by a financial one in the collaborative environment.

In this paper a system structure of ABC/CPPS for semiconductor backend is proposed. After combining the cost data with an ABC model constructed by Pr/Tr Net, a computer-based tool was established to simulate the resources consuming process with the dynamic characteristics in the production line considered. Based on this system structure, costs of the released customer orders (COs) can be rolled up and the MNP can be estimated and reported to the production planner. Furthermore, the collaborative production schedule for each partner to cooperate can be created and published before the COs release to production. Though the MNP we defined in this study does not include all the overhead cost incurred in an enterprise, but a relative MNP index is enough to evaluate the impacts of release or dispatching rules. Other previous research (Ong 1995) shared this same point of view.

Some studies established that a Pr/Tr Net model could be transferred to a rule-based expert system (Giordana and Saitta 1987, Murata and Zhang 1988). In this study, a structured top-down analysis and bottom-up implementation approach is proposed, after implementing the ABC Pr/Tr net into an expert scheduling system by Visual Prolog, it seems useful to generate a collaborative schedule by exploring the alternative dispatching rules on a superior profit basis. If compares to most other tools which are developed base on simulation-based technology, the expert scheduling system is characterized by the following advantages.

- Most of the simulation-based approach using "what-if" analysis procedure, which is long time consuming. The system developed in this study is local optimized by automatically searching into the solution space; therefore less human intervention is required.
- The system can provide as a tool for both in an autonomous control environment or collaborative environment. In addition to an autonomous control production schedule can be created, furthermore, the collaborative schedule for each collaborative partner to cooperate can be planned in the second stage.
- The system provides a planning environment for multi-company, multi-plant and multi-product.
- However, the financial measure, profit, is considered to be more meaningful in a collaborative operation environment. By applying ABC concept to this system, makes the profit information available. Meanwhile, during the process of investigating into the cost of each activity (operation), opportunities of cost down, which enhance the overall competitive advantage, might appear.
- Goals that improve or empower the system are easily to be incorporated and adapted.

- Expert knowledge is easily data publish, to be included to make the system more intelligent.
- Connect to web site for easily data exchange, publish and communication.

Finally, we have to emphasize, though the investment in semiconductor backend industry is far less than in the frontend, firms in the semiconductor backend are vital and critical in the supply chain. Thus, more studies are needed since backend firm is harder to survive.

7.2 Future Study

In addition to an example case is used to illustrate the potential application capability of ABC/CPPS in this study, however, it appears as a useful tool in following future research topics:

- Release policy: Some studies indicated that the order release policy is the dominant factor in determining most of the production system performance (Ragatz and Mabert 1988, Wein 1988). And most semiconductor firms use EDD to release customer orders, but usually the consequence on profit is unknown. Therefore, a study about release policy is needed for practical purposes. The ABC/CPPS system structure proposed in this study provides a tool to investigate this topic.
- Dispatching rule: The dispatching rule used on the shop floor impacts the system performance secondary. Therefore, except the EDD, what other dispatching rule results better profit can be explored in the future.
- Long term planning: Though Bakke and Hellberg (2002) and Kee and Schmid (2000) found that ABC generates higher profits in the long term. But another research conducted by Lea *et al.* (1994) pointed out that ABC generates higher profits for both the short and long term. How time horizon impacts on profit

can be examined.

- Profit sharable analysis: From the collaborative planning/operation, profit might be improved, how to share the profit for each member in a collaborative environment also appears as another interesting study subject.
- Partner alliance: The Holonic Manufacturing System (HMS) concept has been discussed by Huang *et al.* (2002), under the HMS structure, the cooperative partners changes rapidly, thus how to select right ones to collaborate in order to achieve better competence or overall profit, is a challenging topic. The model and methodology proposed in this study might provide a useful approach connect to this subject.

Lastly, in addition to the future topics listed above, a dynamic integrated performance measure system that combines both financial and non-financial factors and fits to the semiconductor industry is worth investigating on a continuous basis. The unique feature of such a system is, according to the planner's weighting on the financial and non-financial factors, the system is adjustable.

APPENDIX A: Collaborative schedule for assembly firm ASE

ruleset: rule("EDD","lcst","nonsubcontract") profit= 895332

Activity	Order No	Company	Plant	Resource type	Resource No.	From	To
1	ao01	ase	plant 1	Production Planner	1	0	2
1	tk01	ase	plant 1	Production Planner	1	2	4
1	tk02	ase	plant 1	Production Planner	1	4	6
1	tk03	ase	plant 1	Production Planner	1	6	8
1	tk04	ase	plant 1	Production Planner	1	8	10
1	tk05	ase	plant 1	Production Planner	1	10	12
1	tk06	ase	plant 1	Production Planner	1	12	14
1	tk07	ase	plant 1	Production Planner	1	14	16
2	ao01	ase	plant 1	Production Planner	1	2	5
2	tk01	ase	plant 1	Production Planner	1	5	8
2	tk02	ase	plant 1	Production Planner	1	8	11
2	tk03	ase	plant 1	Production Planner	1	11	14
2	tk04	ase	plant 1	Production Planner	1	14	17
2	tk05	ase	plant 1	Production Planner	1	17	20
2	tk06	ase	plant 1	Production Planner	1	20	23
2	tk07	ase	plant 1	Production Planner	1	23	26
3	ao01	ase	plant 1	Production Planner	1	5	9
3	tk01	ase	plant 1	Production Planner	1	9	13
3	tk02	ase	plant 1	Production Planner	1	13	17
3	tk03	ase	plant 1	Production Planner	1	17	21
3	tk04	ase	plant 1	Production Planner	1	21	25
3	tk05	ase	plant 1	Production Planner	1	25	29
3	tk06	ase	plant 1	Production Planner	1	29	33
3	tk07	ase	plant 1	Production Planner	1	33	37
4	ao01	ase	plant 1	Production Planner	1	9	11
4	tk01	ase	plant 1	Production Planner	1	13	15
4	tk02	ase	plant 1	Production Planner	1	17	19
4	tk03	ase	plant 1	Production Planner	1	21	23
4	tk04	ase	plant 1	Production Planner	1	25	27
4	tk05	ase	plant 1	Production Planner	1	29	31
4	tk06	ase	plant 1	Production Planner	1	33	35
4	tk07	ase	plant 1	Production Planner	1	37	39
5	ao01	ase	plant 1	Operator	1	11	14
5	tk01	ase	plant 1	Operator	1	15	18
5	tk02	ase	plant 1	Operator	1	19	22
5	tk03	ase	plant 1	Operator	1	23	26
5	tk04	ase	plant 1	Operator	1	27	30
5	tk05	ase	plant 1	Operator	1	31	34
5	tk06	ase	plant 1	Operator	1	35	38
5	tk07	ase	plant 1	Operator	1	39	42
7	ao01	ase	plant 1	Wafer Grinder	3	14	16
7	tk01	ase	plant 1	Wafer Grinder	2	18	20
7	tk02	ase	plant 1	Wafer Grinder	1	22	24
7	tk03	ase	plant 1	Wafer Grinder	3	26	28
7	tk04	ase	plant 1	Wafer Grinder	2	30	32
7	tk05	ase	plant 1	Wafer Grinder	1	34	36
7	tk06	ase	plant 1	Wafer Grinder	3	38	40
7	tk07	ase	plant 1	Wafer Grinder	2	42	44
9	ao01	ase	plant 1	Wafer Saw	3	16	20
9	tk01	ase	plant 1	Wafer Saw	1	20	24
9	tk02	ase	plant 1	Wafer Saw	3	24	28
9	tk03	ase	plant 1	Wafer Saw	1	28	32
9	tk04	ase	plant 1	Wafer Saw	3	32	36

9	tk05	ase	plant 1	Wafer Saw	1	36	40
9	tk06	ase	plant 1	Wafer Saw	3	40	44
9	tk07	ase	plant 1	Wafer Saw	1	44	48
11	ao01	ase	plant 1	Die Bonder	5	20	22
11	tk01	ase	plant 1	Die Bonder	4	24	26
11	tk02	ase	plant 1	Die Bonder	3	28	30
11	tk03	ase	plant 1	Die Bonder	2	32	34
11	tk04	ase	plant 1	Die Bonder	1	36	38
11	tk05	ase	plant 1	Die Bonder	5	40	42
11	tk06	ase	plant 1	Die Bonder	4	44	46
11	tk07	ase	plant 1	Die Bonder	3	48	50
12	ao01	ase	plant 1	Inspector	1	22	25
12	tk01	ase	plant 1	Inspector	1	26	29
12	tk02	ase	plant 1	Inspector	1	30	33
12	tk03	ase	plant 1	Inspector	1	34	37
12	tk04	ase	plant 1	Inspector	1	38	41
12	tk05	ase	plant 1	Inspector	1	42	45
12	tk06	ase	plant 1	Inspector	1	46	49
12	tk07	ase	plant 1	Inspector	1	50	53
14	ao01	ase	plant 1	Wire Bonder	5	25	27
14	tk01	ase	plant 1	Wire Bonder	4	29	31
14	tk02	ase	plant 1	Wire Bonder	2	33	35
14	tk03	ase	plant 1	Wire Bonder	3	37	39
14	tk04	ase	plant 1	Wire Bonder	1	41	43
14	tk05	ase	plant 1	Wire Bonder	5	45	47
14	tk06	ase	plant 1	Wire Bonder	2	49	51
14	tk07	ase	plant 1	Wire Bonder	4	53	55
15	ao01	ase	plant 1	3/O Inspector	1	27	30
15	tk01	ase	plant 1	3/O Inspector	1	31	34
15	tk02	ase	plant 1	3/O Inspector	1	35	38
15	tk03	ase	plant 1	3/O Inspector	1	39	42
15	tk04	ase	plant 1	3/O Inspector	1	43	46
15	tk05	ase	plant 1	3/O Inspector	1	47	50
15	tk06	ase	plant 1	3/O Inspector	1	51	54
15	tk07	ase	plant 1	3/O Inspector	1	55	58
17	ao01	ase	plant 1	Molder	3	30	32
17	tk01	ase	plant 1	Molder	2	34	36
17	tk02	ase	plant 1	Molder	1	38	40
17	tk03	ase	plant 1	Molder	3	42	44
17	tk04	ase	plant 1	Molder	2	46	48
17	tk05	ase	plant 1	Molder	1	50	52
17	tk06	ase	plant 1	Molder	3	54	56
17	tk07	ase	plant 1	Molder	2	58	60
19	ao01	ase	plant 1	Oven	2	32	36
19	tk01	ase	plant 1	Oven	1	36	40
19	tk02	ase	plant 1	Oven	2	40	44
19	tk03	ase	plant 1	Oven	1	44	48
19	tk04	ase	plant 1	Oven	2	48	52
19	tk05	ase	plant 1	Oven	1	52	56
19	tk06	ase	plant 1	Oven	2	56	60
19	tk07	ase	plant 1	Oven	1	60	64
21	ao01	ase	plant 1	Laser Marker	2	36	38
21	tk01	ase	plant 1	Laser Marker	1	40	42
21	tk02	ase	plant 1	Laser Marker	2	44	46
21	tk03	ase	plant 1	Laser Marker	1	48	50
21	tk04	ase	plant 1	Laser Marker	2	52	54
21	tk05	ase	plant 1	Laser Marker	1	56	58
21	tk06	ase	plant 1	Laser Marker	2	60	62

21	tk07	ase	plant 1	Laser Marker	1	64	66
23	ao01	ase	plant 1	Damber Cutter	1	38	42
23	tk01	ase	plant 1	Damber Cutter	1	42	46
23	tk02	ase	plant 1	Damber Cutter	1	46	50
23	tk03	ase	plant 1	Damber Cutter	1	50	54
23	tk04	ase	plant 1	Damber Cutter	1	54	58
23	tk05	ase	plant 1	Damber Cutter	1	58	62
23	tk06	ase	plant 1	Damber Cutter	1	62	66
23	tk07	ase	plant 1	Damber Cutter	1	66	70
25	ao01	ase	plant 1	Solder Plating	2	42	45
25	tk01	ase	plant 1	Solder Plating	1	46	49
25	tk02	ase	plant 1	Solder Plating	2	50	53
25	tk03	ase	plant 1	Solder Plating	1	54	57
25	tk04	ase	plant 1	Solder Plating	2	58	61
25	tk05	ase	plant 1	Solder Plating	1	62	65
25	tk06	ase	plant 1	Solder Plating	2	66	69
25	tk07	ase	plant 1	Solder Plating	1	70	73
27	ao01	ase	plant 1	Trim/Form	1	45	47
27	tk01	ase	plant 1	Trim/Form	1	49	51
27	tk02	ase	plant 1	Trim/Form	1	53	55
27	tk03	ase	plant 1	Trim/Form	1	57	59
27	tk04	ase	plant 1	Trim/Form	1	61	63
27	tk05	ase	plant 1	Trim/Form	1	65	67
27	tk06	ase	plant 1	Trim/Form	1	69	71
27	tk07	ase	plant 1	Trim/Form	1	73	75
28	ao01	ase	plant 1	VI Inspection	1	47	50
28	tk01	ase	plant 1	VI Inspection	1	51	54
28	tk02	ase	plant 1	VI Inspection	1	55	58
28	tk03	ase	plant 1	VI Inspection	1	59	62
28	tk04	ase	plant 1	VI Inspection	1	63	66
28	tk05	ase	plant 1	VI Inspection	1	67	70
28	tk06	ase	plant 1	VI Inspection	1	71	74
28	tk07	ase	plant 1	VI Inspection	1	75	78
29	ao01	ase	plant 1	Paking Operator	1	50	54
29	tk01	ase	plant 1	Paking Operator	1	54	58
29	tk02	ase	plant 1	Paking Operator	1	58	62
29	tk03	ase	plant 1	Paking Operator	1	62	66
29	tk04	ase	plant 1	Paking Operator	1	66	70
29	tk05	ase	plant 1	Paking Operator	1	70	74
29	tk06	ase	plant 1	Paking Operator	1	74	78
29	tk07	ase	plant 1	Paking Operator	1	78	82
30	ao01	ase	plant 1	Ship out Operator	1	54	56

APPENDIX B: Collaborative schedule for assembly firm ASET

ruleset: rule("EDD","lcst","nonsubcontract") profit= 895332

Activity	Order No	Company	Plant	Resource type	Resource No.	From	To
32	tk02	aset	plant 1	Burn/In Oven	1	62	65
32	tk05	aset	plant 1	Burn/In Oven	1	74	77
34	tk01	aset	plant 1	FT1 Tester	3	61	64
34	tk06	aset	plant 1	FT1 Tester	2	81	84
36	tk01	aset	plant 1	FT2 Tester	4	64	68
36	tk02	aset	plant 1	FT2 Tester	3	68	72
36	tk03	aset	plant 1	FT2 Tester	2	72	76
36	tk04	aset	plant 1	FT2 Tester	1	76	80
38	tk01	aset	plant 1	FT3 Tester	3	68	71
38	tk02	aset	plant 1	FT3 Tester	2	72	75
40	tk01	aset	plant 1	Marker	1	71	74
40	tk03	aset	plant 1	Marker	1	79	82
40	tk05	aset	plant 1	Marker	1	87	90
40	tk07	aset	plant 1	Marker	1	95	98
41	tk01	aset	plant 1	EQA	1	74	77
41	tk02	aset	plant 1	EQA	1	78	81
41	tk03	aset	plant 1	EQA	1	82	85
41	tk04	aset	plant 1	EQA	1	86	89
41	tk05	aset	plant 1	EQA	1	90	93
41	tk06	aset	plant 1	EQA	1	94	97
41	tk07	aset	plant 1	EQA	1	98	101
43	tk01	aset	plant 1	Lead Scanner	1	77	81
43	tk02	aset	plant 1	Lead Scanner	1	81	85
43	tk03	aset	plant 1	Lead Scanner	1	85	89
43	tk04	aset	plant 1	Lead Scanner	1	89	93
43	tk05	aset	plant 1	Lead Scanner	1	93	97
43	tk06	aset	plant 1	Lead Scanner	1	97	101
43	tk07	aset	plant 1	Lead Scanner	1	101	105
44	tk01	aset	plant 1	VI Inspector	1	81	84
44	tk02	aset	plant 1	VI Inspector	1	85	88
44	tk03	aset	plant 1	VI Inspector	1	89	92
44	tk04	aset	plant 1	VI Inspector	1	93	96
44	tk05	aset	plant 1	VI Inspector	1	97	100
44	tk06	aset	plant 1	VI Inspector	1	101	104
44	tk07	aset	plant 1	VI Inspector	1	105	108
48	tk01	aset	plant 1	T&R Machine	2	88	91
48	tk02	aset	plant 1	T&R Machine	1	92	95
48	tk04	aset	plant 1	T&R Machine	2	100	103
48	tk05	aset	plant 1	T&R Machine	1	104	107
48	tk07	aset	plant 1	T&R Machine	2	112	115
49	tk01	aset	plant 1	P/K Operator	1	91	95
49	tk03	aset	plant 1	P/K Operator	1	99	103
49	tk05	aset	plant 1	P/K Operator	1	107	111
49	tk07	aset	plant 1	P/K Operator	1	115	119

APPENDIX C: Collaborative schedule for assembly firm WTAE

ruleset: rule("EDD","lcst","nonsubcontract") profit= 895332

Activity	Order No	Company	Plant	Resource type	Resource No.	From	To
32	tk01	wtae	plant 2	Burn/In Oven	2	58	61
32	tk03	wtae	plant 2	Burn/In Oven	1	66	69
32	tk04	wtae	plant 2	Burn/In Oven	2	70	73
32	tk06	wtae	plant 2	Burn/In Oven	1	78	81
32	tk07	wtae	plant 2	Burn/In Oven	2	82	85
34	tk02	wtae	plant 2	FT1 Tester	4	65	68
34	tk03	wtae	plant 2	FT1 Tester	2	69	72
34	tk04	wtae	plant 2	FT1 Tester	5	73	76
34	tk05	wtae	plant 2	FT1 Tester	3	77	80
34	tk07	wtae	plant 2	FT1 Tester	1	85	88
36	tk05	wtae	plant 2	FT2 Tester	5	80	84
36	tk06	wtae	plant 2	FT2 Tester	4	84	88
36	tk07	wtae	plant 2	FT2 Tester	3	88	92
38	tk03	wtae	plant 2	FT3 Tester	5	76	79
38	tk04	wtae	plant 2	FT3 Tester	4	80	83
38	tk05	wtae	plant 2	FT3 Tester	3	84	87
38	tk06	wtae	plant 2	FT3 Tester	2	88	91
38	tk07	wtae	plant 2	FT3 Tester	1	92	95
40	tk02	wtae	plant 2	Marker	1	75	78
40	tk04	wtae	plant 2	Marker	1	83	86
40	tk06	wtae	plant 2	Marker	1	91	94
46	tk01	wtae	plant 2	Oven	1	84	88
46	tk02	wtae	plant 2	Oven	1	88	92
46	tk03	wtae	plant 2	Oven	1	92	96
46	tk04	wtae	plant 2	Oven	1	96	100
46	tk05	wtae	plant 2	Oven	1	100	104
46	tk06	wtae	plant 2	Oven	1	104	108
46	tk07	wtae	plant 2	Oven	1	108	112
48	tk03	wtae	plant 2	T&R Machine	2	96	99
48	tk06	wtae	plant 2	T&R Machine	1	108	111
49	tk02	wtae	plant 2	P/K Operator	1	95	99
49	tk04	wtae	plant 2	P/K Operator	1	103	107
49	tk06	wtae	plant 2	P/K Operator	1	111	115
50	tk01	wtae	plant 2	FQA Inspector	1	95	98
50	tk02	wtae	plant 2	FQA Inspector	1	99	102
50	tk03	wtae	plant 2	FQA Inspector	1	103	106
50	tk04	wtae	plant 2	FQA Inspector	1	107	110
50	tk05	wtae	plant 2	FQA Inspector	1	111	114
50	tk06	wtae	plant 2	FQA Inspector	1	115	118
50	tk07	wtae	plant 2	FQA Inspector	1	119	122
51	tk01	wtae	plant 2	Inventory Storage	1	98	103
51	tk02	wtae	plant 2	Inventory Storage	1	103	108
51	tk03	wtae	plant 2	Inventory Storage	1	108	113
51	tk04	wtae	plant 2	Inventory Storage	1	113	118
51	tk05	wtae	plant 2	Inventory Storage	1	118	123
51	tk06	wtae	plant 2	Inventory Storage	1	123	128
51	tk07	wtae	plant 2	Inventory Storage	1	128	133
52	tk01	wtae	plant 2	Shipping Op	1	103	108
52	tk02	wtae	plant 2	Shipping Op	1	108	113
52	tk03	wtae	plant 2	Shipping Op	1	113	118
52	tk04	wtae	plant 2	Shipping Op	1	118	123
52	tk05	wtae	plant 2	Shipping Op	1	123	128
52	tk06	wtae	plant 2	Shipping Op	1	128	133

52	tk07	wtae	plant 2	Shipping Op	1	133	138

APPENDIX D

IMPLEMENTAITON OF XML DATA EXCHANGE

D.1 XML and Data Exchange Flow

Similar to HTML, XML uses tags to constitute the document contents. In fact, both languages originate from the same parent SGML; thus XML is a subset of SGML. However, there is little difference in tag between the two languages: the tags of HTML are mainly for presenting the contents of web page, but tags for XML are mainly for describing the structure of the contents. By the way, the tags for XML are definable according to one's needs. Therefore, in essence, XML is excellent for data exchange in the Internet

Contrast to Electronic Data Interchange (EDI), XML is an open format with focus on information processing. In addition to the capability of crossing heterogeneous platform, XML data stream can be controlled and displayed the way now IT people manipulates the text and graphics. Moreover, XML is more powerful in its ease of use and customization to user's needs. The cost is also low when compared to EDI, therefore XML is popular for smaller business. Obviously, XML is becoming as the standard and prevailing fast. The advantages of XML are listed as follows (Yen et al. 2002):

- Direct useable through Internet (low cost)
- Support many softwares
- Easy to be processed by program
- Formal and succinct for design
- Easy and fast to be generated
- Define the tags by required

As shown in Figure D.1, collaborative members transfer collaborative production resource data (in XML format) to VE through the Internet, and then resource data are transformed into internal format which can be used by CPPS. After the collaborative production schedule (internal format) is planned out, it is transformed back into XML format (by referring to DTD) for ease of data exchange among partners' systems.

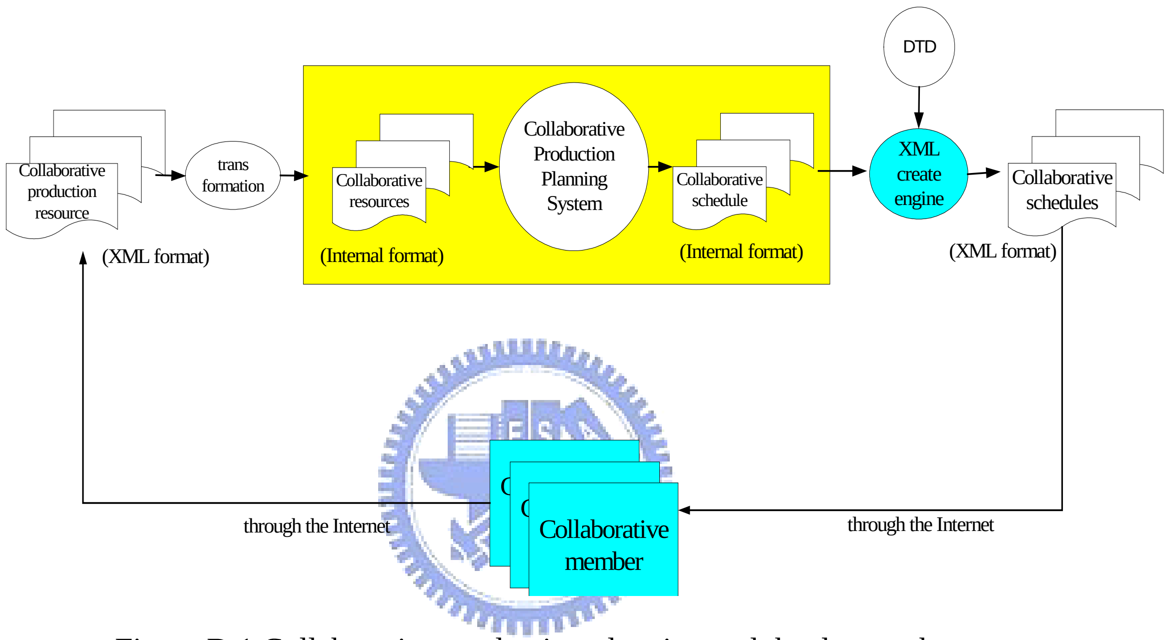


Figure D.1 Collaborative production planning and the data exchange

D.2 DTD

As described previously, one of the advantages of XML is that XML tags are definable; usually tags are defined in DTD (Document Type Definition) and used in XML document. In essence, DTD differs from XML document in that DTD is used to define the tags not the contents of document. Therefore a XML document is called "validated" if tags used in XML document are formally defined in DTD. The common key words used for definitions in an XML document are introduced below (Tseng and Huang , 2002):

- ELEMENT: ELEMENT is used to define an element type, which corresponds to an element of a DI conforming to the element type used in the XML document. Take <!ELEMENT *element_type (content)*> as an example, the element element_type which appears between the symbols ”<!” and ”>” is declared as a tag. On the other hand the keyword “content” defines the data type of the element_type; such as #PCDATA defines the data type as character string.

- ATTLIST: ATTLIST is used to define the attributes and attribute values of an ELEMENT used in an XML document. Take <!ATTLIST *element_type attribute_name type default*> as an example, an attribute attribute_name is declared, *type* is used to define the data type of the attribute, *default* is used for initiating a default value. Ten data types are available in DTD, they are CDDATA, Enumerated, NMTODENS, ENTITY, ENTITIES, ID, IDREF, IDREFS and NOTATION. The last parameter, *default*, is used for defining the characteristic of an attribute; such as #FIXED, #REQUIRED, #IMPLED and #DEFAULT are available string values for parameter *default*. #FIXED is used to define a fixed value, #REQUIRED is used to define an attribute that is necessary, but #IMPLED is used to define an attribute that is optional. Finally, #Default provides a fixed default value. The general expression includes element and attribute is illustrated as *<element_type attribute_name=value>*.

D.3 XML Document

In Figure D.2, a DTD document is illustrated; ELEMENT is used and seven elements (tags), the author, authorlist, book, booklist, code, price and title are defined. The data type of these tags is character string since the content is defined to #PCDATA. The tag, booklist, is constructed by one or more than one book since symbol (+) appears in row <!ELEMENT booklist (book+)> and defined as the content. If expand the tag book, other tags code, title, authorlist and price will appear on the tree structure.

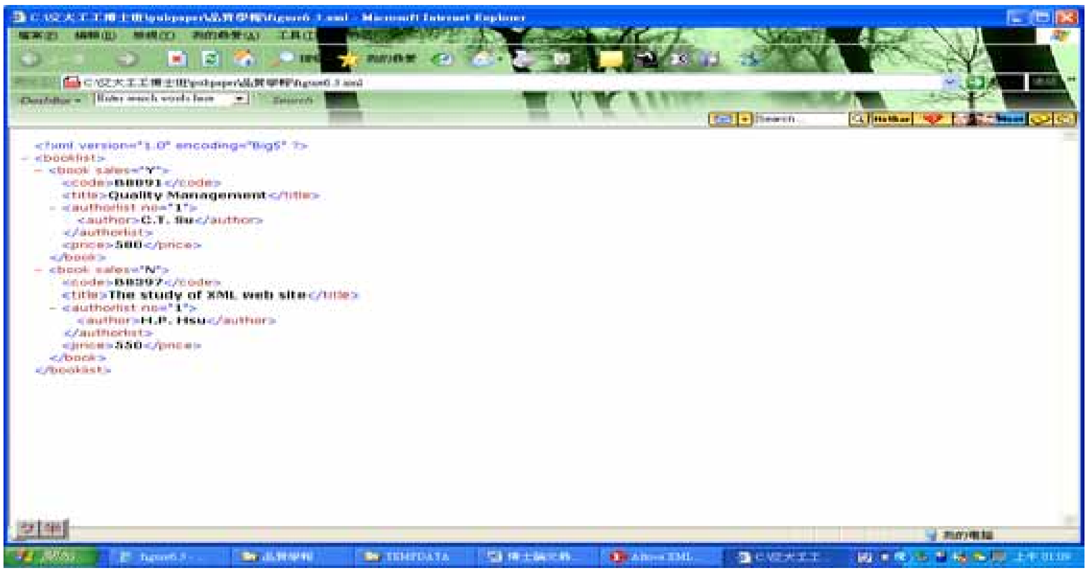
Figure D.3 illustrates an XML document example which includes data items and tag elements, and this document is called “validated” since all tags used in this XML document are all defined in Figure D.2. Figure D.4 displays the view by using IE web browser.

```
<!ELEMENT author (#PCDATA)>
<!ELEMENT authorlist (author)>
<!ATTLIST authorlist no CDATA #REQUIRED>
<!ELEMENT book (code, title, authorlist, price)>
<!ATTLIST book sales (N | Y) #REQUIRED>
<!ELEMENT booklist (book+)>
<!ELEMENT code (#PCDATA)>
<!ELEMENT price (#PCDATA)>
<!ELEMENT title (#PCDATA)>
```

Figure D.2 DTD example

```
<?xml version="1.0" encoding="Big5"?>
<booklist>
    <book sales="Y">
        <code>B8891</code>
        <title>Quality Management</title>
        <authorlist no="1">
          <author>C.T. Su</author>
        </authorlist>
        <price>580</price>
    </book>
    <book sales="N">
        <code>B8397</code>
        <title>The study of XML web site</title>
        <authorlist no="1">
          <author>H.P. Hsu</author>
        </authorlist>
      <price>550</price>
    </book>
 </booklist>
```

Figure D.3 XML document example



```
<?xml version="1.0" encoding="Big5" ?>
- <booklist>
  - <book sales="Y">
      <code>B8091</code>
      <title>Quality Management</title>
    - <authorlist no="1">
        <author>C.T. Su</author>
      </authorlist>
      <price>580</price>
    </book>
  - <book sales="N">
      <code>B8297</code>
      <title>The study of XML web site</title>
    - <authorlist no="1">
        <author>H.P. Hsu</author>
      </authorlist>
      <price>550</price>
    </book>
</booklist>
```

Figure D.4 view displayed by using IE web browser for Figure 6.3

D.4 XML Data Conversion

In this study, the conversion of a collaborative scheduling data from an internal format to XML document is investigated. Three steps to complete the conversion work are proposed as follows:

Step 1: Understanding the data structure used in internal system

Step 2: Design of the DTD used in XML document

Step 3: Development of the XML generating engine

D.4.1 Understanding the Data Structure

Before start designing the DTD for a XML document, analysis of the internal data items must be conducted firstly. Refer to Figure D.1 there are two internal data structures used by ABC/CPPS, the available resources and the resulted production schedule. Available resources are resources released and declared by collaborative members, so they are input data for ABC/CPPS; on the contrary, the resulted collaborative production schedule is the outcome from the ABC/CPPS. This study focuses on the development of the later one.

Basically, the internal data structure varies with system. In this study, the data structure used in Visual Prolog is characterized by Object-Orient concept. As shown by Figure 5.1(d), the object collaborative_schedule which containing data items company_name, plant_name, resource_id, customer, customer_no, operation, schd_from and schd_to is organized as a tree structure.

D.4.2 Design of the DTD Used in XML Document

After analyzing the data items and structure of internal data, the next step is going on to design the DTD. By considering the schema after expanding an XML document and along with the assistance of a specific tool—the abstract tree, DTD can be defined quickly. As shown in Figure D.5, collaborative_schedule is the top (root) element in the abstract tree; the left dotted downward arrow implies that if tag collaborative_schedule is clicked and expanded, there will appear a tag company, and then if company tag is clicked, tag plant will then show up. Therefore, in the same way, if tag plant is clicked and expanded then tag resource will appear. Finally, tag operation will appear if tag resource is expanded and the scheduling data (the span) for each resource will appear. Note another one, if there is a upward dotted arrow pointing to an upper-level tag from a lower-level tag, this implies the upper-level tag may contain multiple direct lower-level tags in this hierarchy relationship; such as in Figure D.5 an upward dotted arrow pointing from company tag to collaborative_schedule tag, thus, under collaborative_schedule tag, there may appear multiple (more than one) company tags when collaborative_schedlue is expanded.

With the help of abstract tree, DTD can be derived quickly. According to the top-down procedure and by using the keyword ELEMENT iteratively, tags corresponding to specific levels in the abstract tree can be derived one by one. For

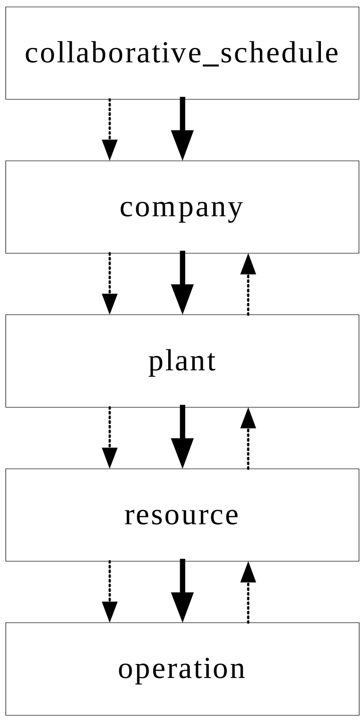


Figure D.5 Abstract tree structure—DTD for collaborative schedule

example, at the top level of the abstract tree there is a root element, collaborative_schedule, and under the root element, there exists company tags which are used to represent distinct company entities collaborating under this schedule. Hence expression <!ELEMENT collaborative_schedule (company+)> is used to define this relationship. Moreover, under a specific company, there might own many different plants, thus <!ELEMENT company (plant+)> is used to represent this relationship. Other contents in the DTD can be defined in the same way, and finally the contents will be completed. A different treatment in the DTD is that schedule span will appear as a data item for tag operation in a character string type; therefore keyword #PCDATA is used, and <!ELEMENT operation (#PCDATA)> is defined in the DTD document. When definitions for each ELEMENT are completed, ATTLIST is used to define the attribute for each tag. Finally the entire DTD (Figure D.6) is then derived and saved to file "collaborativeschedule.dtd" for further invoked by XML document.

```
<!ELEMENT collaborative_schedule (company+)>
<!ELEMENT company (plant+)>
<!ELEMENT plant (resource+)>
<!ELEMENT resource (operation+)>
<!ELEMENT operation (#PCDATA)>
<!ATTLIST company company_name (ASE | ASET|WTAE) #REQUIRED>
<!ATTLIST plantl plant_name CDATA #REQUIRED>
<!ATTLIST resource resource_id CDATA #REQUIRED>
<!ATTLIST operation operation_id CDATA #REQUIRED>
<!ATTLIST operation order_no CDATA #REQUIRED>
```

Figure D.6 The DTD for collaborative schedule

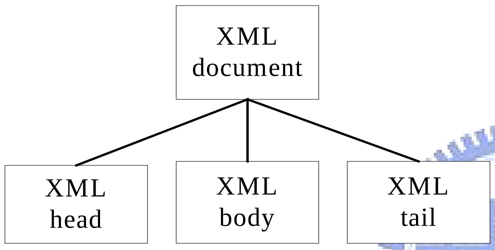
Using the tags defined in DTD file, XML document can be displayed as shown in Figure D.7 if data items from internal system are available and tags used in this document are taken from DTD. Note that the bolded parameters are data values derived from internal system. In line 2, the parameter SYSTEM is used for indicating the use of the source file of DTD (collaborativeschedule.dtd) for validation.

```
<?xml version="1.0" encoding="BIG5" ?>
<!DOCTYPE collaborative_schedule SYSTE M "collaborativeschedule.dtd">
 <collaborative_schedule>
  <company company_name="company_name">
   <plant plant_name="plant_name">
    <resource resource_id="resource_id">
     <operation operation_id="operation_id" order_no="order_no">schd_from-schd_to</operation>
          :
     <operation operation_id="operation_id" order_no="order_no">schd_from-schd_to</operation>
    </resource>
   </plant>
 </company>
  <collaborative_schedule>
  <company company_name="company_name">
   <plant plant_name="plant_name">
    <resource resource_id="resource_id">
     <operation operation_id="operation_id" order_no="order_no">schd_from-schd_to</operation>
          :
     <operation operation_id="operation_id" order_no="order_no">schd_from-schd_to</operation>
    </resource>
   </plant>
 </company>
</collaborative_schedule>
```

Figure D.7 The expected display of XML document

D.4.3 Development of XML document generating engine

The XML document engines is designed for extracting the internal data items from the scheduling system and automatically generate the XML document by using the tags defined in DTD. As shown in Figure D.8, the XML document is mainly constructed by head, body, and tail modules, therefore following this procedure, a program may be designed to generate an XML document in these steps. Corresponding to the reasoning procedure used by Visual Prolog is DFS (Depth First Search), therefore the main functions of the XML generating program can be structured as shown in Figure D.9.



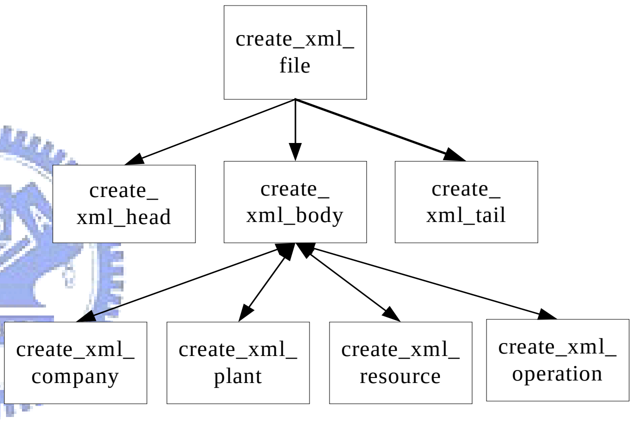


Figure D.8 XML document structure

Figure D.9 Main functions of program

After implementing the main function of the program, a XML generating engine is completed. Partial program of the completed system is shown below. After reading in the data items output from the scheduling system, an XML document is generated and proofed to be a validated and shown in Figure D.10 by IE browser.

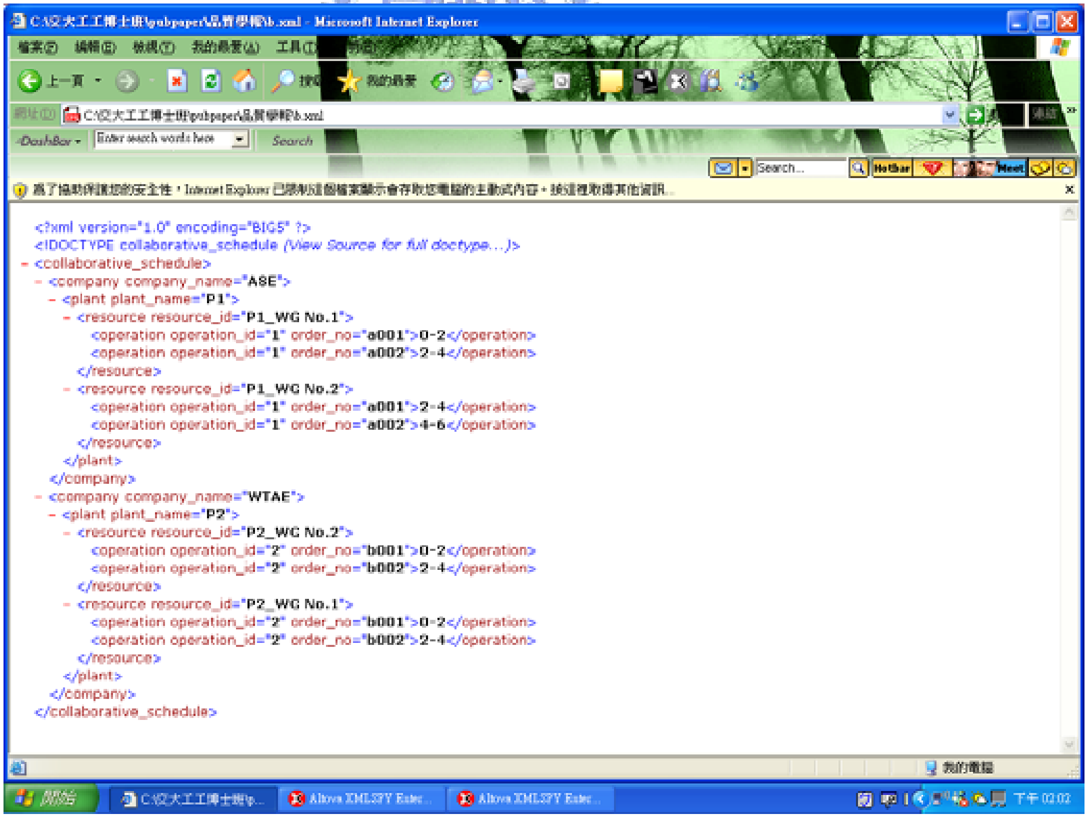
```
create_xml_file():-FileStream = outputStream_file::create8("xmlFile.xml"),
                    create_xml_head(FileStream),
                    create_xml_body(FileStream),
                    create_xml_tail(FileStream).
```

```
create_xml_head(FileStream):-
                writeToStream(FileStream,"<?xml version="),
                writeToStreamquote(FileStream,"1.0"),
                writeToStream(FileStream," encoding="),
                writeToStreamquote(FileStream,"BIG5"),
                writeToStream(FileStream," ?>"),
                FileStream:nl(),
                writeToStream(FileStream,"<collaborative_schedule>"),
                FileStream:nl().

create_xml_body(FileStream):-
                partner(Partner),
                create_xml_company(FileStream,Partner),.
                create_xml_plant(FileStream,Partner),
                create_xml_resource(FileStream,Partner),
                create_xml_operation(FileStream,Partner),
                fail.

create_xml_body(FileStream):-FileStream:nl().

Create_xml_tail(FileStream):-
                writeToStream(FileStream,"</collaborative_schedule>"),
                FileStream:close(),!.
```



```
<?xml version="1.0" encoding="BIG5" ?>
<!DOCTYPE collaborative_schedule (View Source for full doctype...)>
- <collaborative_schedule>
  - <company company_name="ASE">
    - <plant plant_name="P1">
      - <resource resource_id="P1_WG No.1">
          <operation operation_id="1" order_no="a001">0-2</operation>
          <operation operation_id="1" order_no="a002">2-4</operation>
        </resource>
      - <resource resource_id="P1_WG No.2">
          <operation operation_id="1" order_no="a001">2-4</operation>
          <operation operation_id="1" order_no="a002">4-6</operation>
        </resource>
      </plant>
    </company>
  - <company company_name="WTAE">
    - <plant plant_name="P2">
      - <resource resource_id="P2_WG No.2">
          <operation operation_id="2" order_no="b001">0-2</operation>
          <operation operation_id="2" order_no="b002">2-4</operation>
        </resource>
      - <resource resource_id="P2_WG No.1">
          <operation operation_id="2" order_no="b001">0-2</operation>
          <operation operation_id="2" order_no="b002">2-4</operation>
        </resource>
      </plant>
    </company>
  </collaborative_schedule>
```

Figure D.10 collaborative schedule (XMLformat)

REFERENCES

Alla, H. and Ladet, P., 1986, Colored Petri Nets: A tool for model Validation and Simulation of FMS, *Flexible Manufacturing Systems: Methods and Studies, North-Holland*, pp.271-281.

Agliari, A., Diligenti, M. and Zavanella, L., 1995, Variable priority dispatching rules: An analytical approach, *International Journal of Production Economics*, 41, 51-58.

Arzi,Y. and Roll, Y., 1993, Dispatching procedures for a flexible manufacturing cell in constant production circumstance, *International Journal of Operations and Production Management*, 13(11), 35-51.

Bakke, N. and Hellbert, R., 1991, Relevance lost? A critical discussion of different cost accounting principles in connection with decision making for both short and long term production scheduling, *International Journal of Production Economics*, 24, 1-18.

Buzen, J., 1973, Computational algorithms for closed queuing networks with exponential servers, *Communication ACM*, 16, 111-120.

Chan, F.T.S., Chan, H. K. and Kazerooni, A., 2002, A fuzzy multi-criteria decision-making technique for evaluation of scheduling rules, *International Journal of Advanced Manufacturing Technology*, 20, 103-113.

Cooper, R. and Kaplan, R. S., 1988, How cost accounting distorts product costs, *Management Accounting*, 69(10), 20-27.

Cooper, R., 1990, Cost classification in unit-based and activity-based manufacturing cost system, *Journal of Cost Management*, 4(3), 4-14.

Denzler, D.R. and Boe, W.J, 1987, Experimental investigation of Flexible Manufacturing System scheduling decision rules, *International Journal of Production Research*, 25(7), 979-994.

Dhavale, D. G., 1990, A manufacturing cost model for computer-integrated manufacturing systems, *International Journal of Production and Management*, 10(8), 5-18.

Fisher, J., 1992, Use of non-financial performance measures, *Journal of Cost Management*, 6(1), 31-38.

Free, T. and Leachman, C., 1999, Scheduling semiconductor device test operations on multihead testers, *IEEE Transactions on Semiconductor Manufacturing*, 12(4), 523-530.

Foster, G. and Gupta, M., 1990, Activity accounting: An electronics industry implementation, in: R. S. Kaplan (Ed.), *Measures for Manufacturing Excellence*, Harvard Business School Press, MA.

Genrich, H.J. and Lautenbach, K., 1986, System modelling with high-level Petri net, *Theoretical Computer Science*, 13, 109-136.

Genrich, H.J., 1987, Predicate/Transition Nets, *Lecture Notes in Computer Science*, 205, 207-247.

Gentina, J. C. and Corbeel, D, 1987, Colored adaptive structured petri nets: A tool for the automatic synthesis of hierarchical control of FMS, Proceeding of *International Conference of Robotics Automation*,166-1173.

Giordana, A. and L. Saitta, 1885, Modeling Production Rules by Means of Predicate/Transition Networks, *Information Science*, 35,1-41.

Glassey, C. and Resende, M. C., 1988, Closed-loop job release control for VLSI circuit manufacturing, *IEEE Transactions on Semiconductor Manufacturing*, 1, 36-46.

Gordon, W. and G. Newell, 1967, Closed Queuing Systems with Exponential Servers, *Operations Research*, 15, 110-109.

Holthaus, O. and Rajendran, C., 1997, Efficient dispatching rules for scheduling in a job shop, *International Journal of Production Economics*, 48, 87-105.

Huang, B., Gou, H., Liu, W., Li, Y. and Xie, M., 2002, A framework for virtual enterprise control with holonic manufacturing paradigm, *Computers in Industry*, 49, 299-310.

Huang, G. Q. and Mak, K. L., 1999, Design for manufacture and assembly on the Internet, *Computers in Industry*, 38, 17-30.

Hutchison, J., Leoong, K. S. and Ward, D., 1991, Scheduling approaches for random job shop flexible manufacturing systems, *International Journal of Production Research*, 29(5), 1053-1067.

Jackson, J., 1957, Networks of Waiting Lines, *Operations Research*, 5, 1957.

Johri, P. K., 1993. Practical issues in scheduling and dispatching in semiconductor wafer fabrication, *Journal of Manufacturing systems*, l(12), 474-485.

Kalkunte, M.V., 1986, Flexible Manufacturing System: A Review of Modeling approaches for Design, Justification and Operation, *Manufacturing System: Method and Studies*, 3-25.

Kee, R. and Schmidt, C., 2000, A comparative analysis of utilizing activity-based costing and the theory of constraints for making product-mix decisions. *International Journal of Production Economics*, 63, 1-17

Kim, Y. D., Kim, J. U., Lim, S.K.and Jun, H. B., 1998, Due-date based scheduling and control policies in a multi-product semiconductor wafer fabrication facility, *IEEE Transactiosn on Semiconductor Manufacturing*,11(1), 155-164.

Koestler, A., 1989, The Ghost in the Machine, Arkana Books, London, 1989.

Laitinen, E. K., 2002, A dynamic performance measurement system: evidence from small Finnish technology companies, *Scandish Journal of*

Management, 18, 65-99.

Lee, D. Y., Uzsoy, R., Martin, V. A., 1994, Scheduling FMS using Petri net and heuristic search, *IEEE Transactions on Robot, Automation*, 10, 123-132.

Lee, D. Y., Uzsoy, R. and Martin-Vega L.A., 1994, Scheduling FMS using Petri nets and heuristic search, *IEEE Transactions on Robot, Automation*, 10, 123-132.

Liao, D. Y., Chang, S. C., Pei, K. W. and Chang, C. M, 1996, Daily scheduling for R&D semiconductor fabrication, *IEEE Transactions on Semiconductor Manufacturing*, 9(4), 550-561.

Liu, C. Y., and Yih, T.Y., 1995, A framework for part type selection and scheduling in FMS environment, *International Journal of Computer Integrated Manufacturing*, 8, 102-118.

Liu, T. H., Trappey, J. C. and Chan, F. W., 1997, A scheduling system for IC packaging industry using STEP enabling technology, *IEEE Transactions on Component, Packaging and Manufacturing Technology, Part C*, 20(4), 256-267.

Lynch, R. L. and Cross, K. F., 1991, Measure up! Yardsticks for continuous, USA: Blackwell.

Mezgar, I., Kovacs, G.L. and Paganelli, P., 2000, Co-operative production planning for small- and medium- sized enterprises, *International Journal of Production Economics*, 64, 37-48.

Murata, T. and Zhang, D., 1988, A Predicate/Transition Net model for parallel interpretation of logic programming, *IEEE Transactions on Software Engineering*, l14(4), 481-497.

Murata, T., 1989, Petri nets: Properties, Analysis and Application, *Proceedings of the IEEE*, 77(4).

Murata, T. and D. Zhang, 1988, A Predicate/Transition Net Model for Parallel Interpretation of Logic Programming, *IEEE Transactions on Software Engineering*, 14(4), 481-497.

Narahari, Y. and Viswanadham, N, 1985, A Petri net approach to the modeling and analysis of flexible manufacturing, *Annual Operation Research*, 3, 449-472.

Nof, S. Y., 1979, Operational control of item flow in versatile manufacturing systems, *International Journal of Production Research,* 17(5), 479-489.

Ong, N. S., 1995, Manufacturing cost estimation for PCB assembly: An activity-based approach, *International Journal of Production Economics*, 38, 159-172.

Park, T. and Lee, H. , 2001, FMS design model with multiple objectives using compromise programming, *International Journal of Operational Research*, 39(5), 3513-3528.

Pirttila, T. and Hautaniemi, P., 1995, Activity-based costing and distribution logistics management, *International Journal of Production Economics*, 41, 327-333

Ragatz, G. L. and Mabert, V. A., 1988, An evaluation of order release mechanisms in a job-shop environment, *Decision Sciences*, 19, 167-189.

Reyes A., Yu, G. Kelleher, Lloyd, S., 2002, Integrating Petri Nets and hybrid heuristic search for the scheduling of FMS, *Computers in Industry*, 47, 123-128.

Reyes A., Yu, H. and Kelleher, G., 2000, Advanced scheduling methodologies for Flexible Manufacturing System using Petri Nets and heuristic search, *Proceedings of IEEE International Conference on Robotics and Automation*, ICRA200, San Francisco, 24-28.

Salafatinos, C., 1996, Modelling resource supply and demand: Expanding the utility of ABC, *International Journal of Production Economics*, 43, 47-57.

Shukla, C. S. and Chen, F. F., 1996, The state-of-the-art in intelligent real-time FMS control: a comprehensive survey", *Journal of Intelligent Manufacturing*, 7, 441-455.

Solberg, J. R., 1977, An overview of evaluative models for Flexible Manufacturing Systems, *4th International Conference on Production Research*, Tokyo.

Solberg, J. R., 1977, A Mathematical Model of Computerized Manufacturing Systems, *4th International Conference on Production Research*, Tokyo.

Spearman, M. L., Woodruff, D.L., Hopp, W. J., 1990, CONWIP: A pull alternative to kanban, *International Journal of Production Research*, 28, 879-894.

Steck, K. E. and Kem, I., 1991, A flexible approach to part type selection in flexible flow systems using part mix ratios, *International Journal of Production Research*, 29, 53-75.

Stecke, K. E. and Solberg, J. J., 1981, Loading and control policies for a Flexible Manufacturing System, *International Journal of Production Research*, 19(5). 481-490.

Suri R., 1983, An overview of evaluative models for Flexible Manufacturing Systems, *Annuals of Operations Research*, 8-15.

Suri, R. and Hildbrant, R. R., 1984, Modeling Flexible Manufacturing Systems using Mean-Value Analysis, *Journal of Manufacturing Systems*, 3(1), 27-38.

Tseng, F. S. C. and Huang, W. J., 2002, An automatic load/extract scheme for XML documents through object-relational repositories, *The Journal of System and Software*, 64, 207-218.

Tsai, W. H., 1996, A technical note on using work sampling to estimate the effect on activities under activity-based costing, *International Journal of Production Research*, 43, 11-16.

Uzsoy, R. and Lee, C. Y., 1992, A review of production planning and scheduling models in the semiconductor industry. Part I: System characteristics, performance evaluation and production planning, *IIE Transactions*, 24, 47-60.

Valette, R. and Courvoisier, M., 1985, Putting Petri nets to work for controlling flexible manufacturing systems, in *Proceeding of International Conference on Circuit System*, Kyoto, Japan, 929-932.

Van, L., 1986, A Review of Planning Models, *Modelling and Design of Flexible Manufacturing System*, 3-31.

Wein, L. M., 1988, Scheduling semiconductor wafer fabrication, *IEEE Transactions on Semiconductor Manufacturing*, 1(3), 115-130.

Yang, J. and Chang, T. S. 1998, Multiobjective scheduling for IC sort and test with a simulation testbed, *IEEE Transactions on Semiconductor Manufacturing*, 11(2), 304-315.

Yen, D. C., Huang, S. M. and Ku, C. Y., 2002, The impact and implementation of XML on business-to-business commerce, *Computer Standards & Interfaces*, 24, 347-362.